

Université de Montréal

**AI for Molecule Discovery with Multi-Modal
Knowledge**

par

Shengchao Liu

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée en vue de l'obtention du grade de
Philosophiæ Doctor (Ph.D.)
en Discipline

July 25, 2023

Université de Montréal

Faculté des arts et des sciences

Cette thèse intitulée

AI for Molecule Discovery with Multi-Modal Knowledge

présentée par

Shengchao Liu

a été évaluée par un jury composé des personnes suivantes :

Guillaume Rabusseau

(président-rapporteur)

Jian Tang

(directeur de recherche)

Pierre-Louis Bellec

(membre du jury)

Yifeng Li

(examineur externe)

Jacques Bélair

(représentant du doyen de la FESP)

Résumé

L'intelligence artificielle pour la découverte de médicaments a suscité un intérêt croissant pour les communautés de l'apprentissage automatique et de la chimie et de la biologie. Au cours de mes 3 ans de doctorat. recherche, je me suis consacré à l'étude de la modélisation multimodale des molécules, y compris, mais sans s'y limiter, la représentation topologique 2D des molécules, la représentation géométrique 3D, l'apprentissage auto-supervisé, l'apprentissage multi-tâches, la génération structurée (contrôlable) et la dynamique d'apprentissage.

Au cours des six derniers mois (de novembre 2022 à avril 2023), avec le succès de ChatGPT et GPT-4, davantage d'efforts ont été déployés dans le grand modèle de langue (modèle de base AKA). Cela correspond parfaitement à ma direction de recherche, qui vise à combiner plusieurs modalités de molécules pour permettre une adaptation rapide à diverses tâches en aval spécifiques à une tâche.

Dans cette thèse, je voudrais fournir une telle perspective pour la découverte de molécules. Plus précisément, je montrerai comment l'intégration de plusieurs modalités peut améliorer les performances des systèmes d'IA dans la découverte de molécules. Ma recherche vise à contribuer au développement d'un nouveau modèle de base pour la découverte efficace et efficiente de médicaments.

Mots clés: découverte de molécules, topologie en 2D, géométrie en 3D, annotation textuelle, graphe de connaissances biologiques, multimodal, pré-entraînement

Abstract

Artificial intelligence for drug discovery has been revoking an increasing interest in the machine learning and chemistry & biology communities. During my 3-year Ph.D. research, I have devoted myself to studying the multi-modal modeling of molecules, including but not limited to molecule 2D topological representation, 3D geometric representation, self-supervised learning, multi-task learning, (controllable) structured generation, and physics-informed dynamic system.

Additionally, in the past six months, with the success of ChatGPT and GPT-4, more efforts have been put into the large language model (AKA foundation model). This aligns well with my research direction, which aims to combine multiple modalities to enable quick adaptation to various task-specific molecule tasks, such as zero-shot molecule optimization and zero-shot property prediction.

In this thesis, I would like to provide a new perspective on molecule discovery. Specifically, I will showcase how the integration of multiple modalities and advanced representation learning techniques can improve the performance and capability of AI systems in molecule discovery, targeting more realistic and challenging problems. My research seeks to contribute to the development of a novel foundation model for effective and efficient drug discovery.

Keywords: molecule discovery, 2D topology, 3D geometry, textual annotation, biological knowledge graph, multi-modal, pretraining

Contents

Résumé	5
Abstract	7
List of Tables	17
List of Figures	21
Liste des sigles et des abréviations	23
Remerciements	25
Chapter 1. Introduction	27
First Article. Pre-training Molecular Graph Representation with 3D Geometry	31
1. Introduction	32
2. Preliminaries	34
3. GraphMVP: Graph Multi-View Pre-training	35
3.1. Overview of GraphMVP	35
3.2. Contrastive Self-Supervised Learning between 2D and 3D Views	36
3.3. Generative Self-Supervised Learning between 2D and 3D Views	37
3.4. Multi-task Objective Function	39
4. Experiments and Results	39
4.1. Experimental Settings	39
4.2. Main Results on Molecular Property Prediction	40
4.3. Ablation Study: The Effect of Masking Ratio and Number of Conformers ...	40
4.4. Ablation Study: The Effect of Objective Function	42
4.5. Broader Range of Downstream Tasks	43
4.6. Case Study	43
5. Theoretical Insights	44

5.1. Maximizing Mutual Information	44
5.2. 3D Geometry as Privileged Information	45
6. Conclusion and Future Work	45
Second Article. Molecular Geometry Pretraining with SE(3)-Invariant Denoising Distance Matching	
	47
1. Introduction	48
2. Related Work	50
2.1. Equivariant Geometric Molecule Representation Learning	50
2.2. Self-Supervised Learning for Molecule Representation Learning	50
3. Preliminaries	51
4. Method	52
4.1. Coordinate Perturbation for Geometric Data	52
4.2. Coordinate Denoising with MI Maximization Framework: GeoSSL	53
4.3. From Coordinate Denoising to Distance Denoising: GeoSSL-DDM	54
4.3.1. Denoising Distance Matching	55
4.3.2. SE(3)-Invariant Score Network Modeling	56
4.4. Ultimate Objective	56
5. Experiments	57
5.1. Backbone Models	57
5.2. Baselines and Pretraining Dataset	58
5.3. Downstream Tasks on Quantum Mechanics and Force Prediction	59
5.4. Downstream Tasks on Binding Affinity Prediction	60
5.5. Discussion: Connection with Multi-task Pretraining	60
6. Conclusions and Future Directions	61
Third Article. A Group Symmetric Stochastic Differential Equation Model for Molecule Multi-modal Pretraining	
	63
1. Introduction	64
2. Related Work	67
3. Preliminaries	67
4. The MoleculeSDE Method	68

4.1.	An Overview from Mutual Information Perspective	69
4.2.	An SE(3)-Equivariant Conformation Generation	69
4.3.	An SE(3)-Invariant Topology Generation	71
4.4.	Learning and Inference of MoleculeSDE.....	72
4.5.	Theoretical Insights of MoleculeSDE.....	72
5.	Experiments.....	73
5.1.	Pretraining and Baselines.....	73
5.2.	Downstream with 2D Topology.....	75
5.3.	Downstream with 3D Conformation.....	75
5.4.	Downstream with Topology to Conformation.....	75
5.5.	Discussion on MoleculeSDE.....	77
6.	Conclusion and Outlook.....	77
Fourth Article. Structured Multi-task Learning for Molecular Property		
	Prediction.....	79
1.	Introduction.....	80
2.	Related Work.....	82
3.	Problem Definition & Preliminaries.....	82
3.1.	Problem Definition	82
3.2.	Preliminaries	83
4.	Dataset with Explicit Task Relation	84
5.	Method: Structured Task Modeling.....	85
5.1.	Overview	85
5.2.	Modeling Task Relation in Latent Space	86
5.3.	Modeling Task Relation in Output Space	87
5.4.	SGNN-EBM.....	88
5.4.1.	Learning	89
5.4.2.	Inference	90
6.	Experiment Results.....	90
6.1.	Main Results	90
6.2.	Ablation Study 1: The Effect of p_n	92
6.3.	Ablation Study 2: The Effect of L	92

7. Conclusion and Future Direction	93
Fifth Article. Multi-modal Molecule Structure-text Model for Text-based Editing and Retrieval	95
1. Introduction	96
2. Results	99
2.1. Overview and Preliminaries	99
2.2. Two Principles for Downstream Task Design	100
2.3. Downstream: Zero-shot Structure-text Retrieval	102
2.4. Downstream: Zero-shot Text-based Molecule Editing	106
2.5. Downstream: Molecular Property Prediction	107
3. Discussion	108
4. Methods	109
4.1. MoleculeSTM Pretraining	109
4.2. Downstream: Zero-shot Structure-text Retrieval	110
4.3. Downstream: Zero-shot Text-based Molecule Editing	111
Chapter 2. Conclusion	113
Bibliography	115
Appendix A. Appendix for GraphMVP: Pre-training Molecular Graph Representation with 3D Geometry	139
A.1. Self-Supervised Learning on Molecular Graph	139
A.1.1. Contrastive graph SSL	139
A.1.2. Generative graph SSL	140
A.1.3. Predictive graph SSL	140
A.2. Molecular Graph Representation	140
A.2.1. 2D Molecular Graph Neural Network	141
A.2.2. 3D Molecular Graph Neural Network	142
A.2.3. Summary	143
A.3. Maximize Mutual Information	144
A.3.1. Formulation	144
A.3.2. A Lower Bound to MI	145

A.4.	Contrastive Self-Supervised Learning	146
A.4.1.	InfoNCE	146
A.4.2.	EBM-NCE	147
A.4.2.1.	Energy-Based Model (EBM)	147
A.4.2.2.	EBM for MI.....	147
A.4.2.3.	Derivation of conditional EBM with NCE	147
A.4.3.	EBM-NCE v.s. JSE and InfoNCE	149
A.5.	Generative Self-Supervised Learning	151
A.5.1.	Variational Molecule Reconstruction	151
A.5.2.	Variational Representation Reconstruction	152
A.5.3.	Variational Representation Reconstruction and Non-Contrastive SSL	153
A.6.	Dataset Overview	154
A.6.1.	Pre-Training Dataset Overview	154
A.6.2.	Downstream Dataset Overview	155
A.7.	Experiments Details	156
A.7.1.	Self-supervised Learning Baselines	156
A.7.2.	Ablation Study: The Effect of Masking Ratio and Number of Conformers	157
A.7.3.	Ablation Study: Effect of Each Loss Component	158
A.7.4.	Broader Range of Downstream Tasks: Molecular Property Prediction Prediction	158
A.7.5.	Broader Range of Downstream Tasks: Drug-Target Affinity Prediction	159
A.7.6.	Case Studies	159
Appendix B. Appendix for GeoSSL: Molecular Geometry Pretraining with SE(3)-Invariant Denoising Distance Matching.....		163
B.1.	Benchmarks and Related Work	163
B.1.1.	Geometric Neural Networks	163
B.1.2.	Benchmark on QM9	164
B.1.3.	Related Work	164
B.2.	An Example On The Importance of Atom Coordinates	165
B.3.	Mutual Information Maximization with Energy-Based Model	167
B.3.1.	An EBM framework for MI estimation	168
B.3.2.	EBM-NCE for MI estimation	169

B.3.3.	EBM-SM for MI estimation: GeoSSL-DDM.....	169
B.3.4.	Discussions.....	172
B.4.	Experiments.....	173
B.4.1.	Computational Resources.....	173
B.4.2.	Dataset: QM9.....	173
B.4.3.	Dataset: MD17.....	173
B.4.4.	Dataset: LBA & LEP.....	174
B.4.5.	Hyperparameter Specification.....	174
B.4.6.	SchNet as Backbone Model.....	176
B.5.	Ablation Studies.....	177
B.5.1.	The Effect of Annealing Factor in GeoSSL-DDM.....	177
B.5.2.	The Effect on the Number of Noise Layers in GeoSSL-DDM.....	177
B.6.	Strong Model Robustness with Random Seeds.....	179
B.6.1.	PaiNN.....	179
B.6.2.	SchNet.....	180
B.7.	Comparison with a Parallel Work.....	181
Appendix C. Appendix for MoleculeSDE: A Group Symmetric Stochastic		
Differential Equation Model for Molecule Multi-modal		
Pretraining..... 183		
C.1.	Comparison to Related Works.....	183
C.2.	Group Symmetry and Local Frame.....	184
C.2.1.	SE(3)/E(3) Group action and representations.....	184
C.2.2.	Equivariant Frames.....	184
C.3.	Denoising Score Matching.....	185
C.3.1.	Energy-Based Model (EBM).....	185
C.3.2.	Score Matching.....	186
C.3.3.	Proof of DSM.....	188
C.4.	Diffusion Model.....	189
C.4.1.	Pipeline of Denoising Diffusion Probabilistic Model.....	189
C.4.2.	Important Tricks.....	191
C.5.	Stochastic Differential Equation.....	192

C.5.1.	Review of NCSN and DDPM	192
C.5.2.	Stochastic Differential Equation	193
C.5.3.	Stochastic Differential Equation and Score Matching	193
C.6.	Mutual Information and Equivalent Conditional Likelihoods	194
C.6.1.	Variational Representation Reconstruction	195
C.7.	Implementation Details of MoleculeSDE	195
C.7.1.	Backbone Models	195
C.7.2.	Molecule Featurization	195
C.7.3.	Pretraining Hyperparameters	196
C.7.4.	SE(3)-Equivariant SDE Model: From Topology to Conformation	196
C.7.5.	SE(3)-Invariant SDE Model: From Conformation to Topology	197
C.8.	Ablation Studies	198
C.8.1.	Ablation Study on Generative SSL Pretraining	198
C.8.2.	Ablation Study on Atom Features and Comparison with Conformation Generation Methods	199
C.8.3.	Ablation Study on The Effect of Contrastive Learning in MoleculeSDE	201
C.8.4.	PaiNN as Backbone	202
C.8.5.	Quality on Conformation Generation	203
C.9.	Computational Cost on Pretraining	203
Appendix D. Appendix for SGNN-EBM: Structured Multi-task Learning for Molecular Property Prediction		
D.1.	ChEMBL-STRING Dataset Generation	206
D.1.1.	Filtering molecules	206
D.1.2.	Querying the PPI scores	206
D.1.3.	Constructing the Task Relation Graph	207
D.2.	GIN for Molecule Embedding	207
D.3.	GCN for Task Embedding	208
D.4.	SGNN for Modeling Latent Space	208
D.5.	Training Details	209
D.5.1.	Hyperparameter Tuning	209
D.6.	Noise Contrastive Estimation with Energy Tilting Term	210

Appendix E. Appendix for MoleculeSTM: Multi-modal Molecule Structure-text Model for Text-based Editing and Retrieval.....	211
E.1. Pretraining.....	211
E.1.1. PubChemSTM Construction.....	211
E.1.2. Architecture Details.....	214
E.1.3. Pretraining Details.....	214
E.2. Design Principles for Downstream Tasks.....	216
E.3. Downstream: Zero-shot Structure-text Retrieval.....	217
E.3.1. Dataset Construction.....	217
E.3.2. Experiments.....	219
E.3.3. Ablation Study: Fixed Pretrained Encoders.....	220
E.4. Downstream: Zero-shot Text-based Molecule Editing.....	221
E.4.1. Experiment Set-up.....	221
E.4.2. Single-objective Molecule Editing.....	223
E.4.3. Multi-objective Molecule Editing.....	226
E.4.4. Binding-affinity-based Molecule Editing.....	228
E.4.5. Drug Relevance Editing.....	232
E.4.6. Case Studies on Neighborhood Searching for Patent Drug Molecules.....	232
E.5. Downstream: Molecular Property Prediction.....	234

List of Tables

1	GraphMVP results on MoleculeNet	41
2	GraphMVP ablation study on masking ratio	41
3	GraphMVP ablation study on number of conformers	41
4	GraphMVP ablation study on objective function	42
5	GraphMVP results on regression tasks	43
6	GeoSSL results on QM9 with PaiNN	59
7	GeoSSL results on MD17 with PaiNN	59
8	GeoSSL results on LBA & LEP with PaiNN	61
9	MoleculeSDE performance comparison with merely generative pretraining	66
10	MoleculeSDE results on MoleculeNet	74
11	MoleculeSDE results on QM9 with SchNet	74
12	MoleculeSDE results on MD17 with SchNet	74
13	MoleculeSDE results on MoleculeNet with generated conformation	76
14	ChEMBL-STRING statistics	84
15	SGNN-EBM results on ChEMBL-STRING	91
16	SGNN-EBM ablation study on noise distributions	91
17	SGNN-EBM ablation study on the layer number	92
18	MoleculeSTM results on MoleculeNet	108
19	GraphMVP comparison with literature	140
20	GraphMVP 3D GNN reproduced results on QM9	143
21	GraphMVP summary for datasets	155
22	GraphMVP ablation study on masking ratio (full results)	157
23	GraphMVP ablation study on number of conformers (full results)	157
24	GraphMVP ablation study on objective function (full results)	158

25	GraphMVP results on four regression tasks	158
26	GraphMVP results on two binding affinity tasks	159
27	GraphMVP results on spatial diameter prediction	159
28	GraphMVP results on donor-acceptor prediction	160
29	GeoSSL benchmark results on QM9	164
30	GeoSSL toy example on the importance of coordinates	165
31	GeoSSL statistics on MD17	174
32	GeoSSL statistics on LBA & LEP	174
33	GeoSSL hyperparameters	175
34	GeoSSL results on QM9 with SchNet	176
35	GeoSSL results on MD17 with SchNet	176
36	GeoSSL results on LBA & LEP with SchNet	177
37	GeoSSL ablation study on annealing factor	177
38	GeoSSL ablation study on noise layer	178
39	GeoSSL random-seed results on QM9 with PaiNN	179
40	GeoSSL random-seed results on MD17 with PaiNN	179
41	GeoSSL random-seed results on QM9 with SchNet	180
42	GeoSSL random-seed results on MD17 with SchNet	180
43	MoleculeSDE comparison with literature	183
44	MoleculeSDE featurization	196
45	MoleculeSDE hyperparameters	196
46	MoleculeSDE ablation study on generative SSL comparison (MoleculeNet)	198
47	MoleculeSDE ablation study on generative SSL comparison (QM9)	198
48	MoleculeSDE ablation study on generative SSL comparison (MD17)	199
49	MoleculeSDE ablation study on rich features	199
50	MoleculeSDE ablation study on coefficient α_1 (MoleculeNet)	201
51	MoleculeSDE ablation study on coefficient α_1 (QM9)	201
52	MoleculeSDE ablation study on coefficient α_1 (MD17)	201
53	MoleculeSDE results on QM9 with PaiNN	202

54	MoleculeSDE results on MD17 with PaiNN	202
55	MoleculeSDE results on QM9 conformation generation	203
56	MoleculeSDE computational time	204
57	SGNN-EBM hyperparameters	210
58	MoleculeSTM examples of PubChemSTM	213
59	MoleculeSTM vocabulary comparison	213
60	MoleculeSTM model specifications	214
61	MoleculeSTM hyperparameters	215
62	MoleculeSTM statistics on three fields in DrugBank	218
63	MoleculeSTM results on DrugBank-Description with fine-tuning	219
64	MoleculeSTM results on DrugBank-Pharmacodynamics with fine-tuning	219
65	MoleculeSTM results on DrugBank-ATC with fine-tuning	219
66	MoleculeSTM results on DrugBank-Description with linear-probing	220
67	MoleculeSTM results on DrugBank-Pharmacodynamics with linear-probing	220
68	MoleculeSTM results on DrugBank-ATC with linear-probing	220
69	MoleculeSTM results on single-objective molecule editing	223
70	MoleculeSTM analysis on solubility	224
71	MoleculeSTM analysis on permeability	224
72	MoleculeSTM analysis on HBA and HBD	225
73	MoleculeSTM results on multi-objective molecule editing	226
74	MoleculeSTM analysis on multi-objective editing (solubility & HBA/HBD)	227
75	MoleculeSTM analysis on multi-objective editing (solubility & permeability)	227
76	MoleculeSTM ChEMBL assay descriptions	228
77	MoleculeSTM results on binding-affinity-based molecule editing	229
78	MoleculeSTM results on drug relevance editing	232
79	MoleculeSTM analysis on drug relevance editing	233
80	MoleculeSTM statistics on MoleculeNet	234
81	MoleculeSTM hyperparameters on MoleculeNet	235

List of Figures

1	An overview on molecule representation	28
2	GraphMVP pipeline	35
3	GraphMVP ablation study on 3D properties	44
4	GeoSSL illustration on coordinate geometry	49
5	GeoSSL-DDM pipeline	54
6	MoleculeSDE pipeline	65
7	MoleculeSDE illustration three downstream tasks	76
8	SGNN-EBM pipeline	85
9	MoleculeSTM pipeline	98
10	MoleculeSTM results for zero-shot retrieval tasks	101
11	MoleculeSTM results for zero-shot text-based editing	104
12	MoleculeSTM analysis on text-based molecule editing	105
13	Roadmap for future direction	114
14	GraphMVP venn diagram of mutual information	144
15	GraphMVP contrastive SSL pipeline	146
16	GraphMVP VRR SSL pipeline	152
17	GraphMVP statistics on conformers	154
18	GraphMVP occurrence weights for top major conformers	154
19	GraphMVP molecule selection	159
20	GraphMVP successful molecule examples on diameter detection	161
21	GeoSSL pipeline	168
22	GeoSSL-DDM pipeline	169
23	MoleculeSDE pipeline for SE(3)-equivariant 2D-3D SDE model	197

24	MoleculeSDE pipeline for SE(3)-invariant 3D-2D SDE model.....	198
25	SGNN-EBM GNN pipeline	208
26	MoleculeSTM binding-affinity-based molecule editing pipeline	229
27	MoleculeSTM analysis on binding-affinity-based molecule editing.....	231

Liste des sigles et des abréviations

AI	Artificial Intelligence
ML	Machine Learning
MI	Mutual Information
EBM	Energy-based Model
NCE	Noise Contrastive Estimation
CD	Contrastive Divergence
SM	Score Matching
DSM	Denoising Score Matching
GNN	Graph Neural Network
GeoSSL	Geometric Self-Supervised Learning
DDM	Denoising Distance Matching

GraphMVP	Graph Multi-View Pretraining
SDE	Stochastic Differential Equation
MoleculeSTM	Molecule Structure-Text Model
SGNN-EBM	State Graph Neural Network - Energy-based Model
HBA	Hydrogen Bond Acceptor
HBD	Hydrogen Bond Donor

Remerciements

First and foremost, I would like to express my gratitude to my family for their unwavering support.

To my father, Mr. Xiaojun Liu, who passed away three years ago: your absence is still deeply felt in my life, and I often think about you. Your words of wisdom about being a responsible and practical person, and your concern for those suffering from various illnesses, have inspired me to pursue research in AI for drug discovery. Thank you for illuminating my career path.

To my mother, Ms. Huiying Zhu: you have shown incredible strength and support over the past three years. It brings me great joy to have you by my side, and I wish you good health and all the best. I look forward to showing you around the city where I lived during my Ph.D. studies when you visit Montreal next month.

Next, I am deeply thankful to Prof. Jian Tang and our research group for their invaluable support. Despite the challenges posed by COVID-19 and the quarantine, our talented team continues to collaborate remotely and generate great ideas. I look forward to staying in touch and exploring future collaborations.

I have also had the pleasure of meeting research professors and friends who share my passion for AI in drug discovery, and who have supported me in various ways. I would like to express my gratitude to Prof. Hongyu Guo for his insightful discussions and guidance on my research projects and career path. Let's continue to push the foundation model for molecule discovery forward. I would also like to thank Prof. Anima Anandkumar for her visionary proposal and kind support. Working with you has been a pleasure, and I hope to continue exploring the potential of the foundation model for drug discovery, with broader impacts.

I would also like to thank my master's advisors. Prof. Anthony Gitter, thank you for introducing me to the world of AI for drug discovery research. Prof. Yingyu Liang, your strong background in machine learning has been invaluable. Prof. Dimitris Papailiopoulos, thank you for your helpful suggestions on my learning dynamics project.

Pierre-André Noël and David Vazquez, your mentorship during my internship at ServiceNow was truly valuable, and the experience was a delightful journey.

Chengpeng Wang, we have known each other for over ten years, and it has been a pleasure to have your domain expertise in supporting the GraphCG and MoleculeSTM works. I look forward to meeting you in person soon in California.

Yan Ai and Jeanne Luo, thank you for being nice friends to me during the hard times. I wish all of us could have a pleasant future.

Finally, I would like to express my deep appreciation for all the external collaborators who have supported me. I could not have completed my research projects without your help. Thank you to Weitao Du, Qi Liu, Weiyang Liu, Hanchen Wang, Chaowei Xiao, Weili Nie, Zhuoxinran Li, Yutao Zhu, Zhao Xu. Wish you all the best.

Chapter 1

Introduction

My main research interest is to explore artificial intelligence (AI) for molecule discovery, especially in incorporating multi-modal knowledge into the molecule discovery pipeline. AI for molecule discovery has raised increasing interest in both the machine learning (ML) and computational biology & chemistry communities. The main breakthroughs include not limited to virtual screening [198], lead optimization [72, 113, 152], protein folding and inverse folding [94, 115]. Among these tasks, molecule representation learning remains to be the most crucial component in realizing the foundation model for molecule discovery. In what follows, I will briefly introduce the current progress in this research direction and how I proceed it using the multi-modal knowledge.

Molecule Discovery with Multiple Modalities. First, I would like to discuss the six most widely used modalities of molecules, as shown in Figure 1. From a high-level point of view, the modalities can be divided into two big venues: (1) The *internal* view is about the molecule’s chemical structure, including topology and geometry. (2) The *external* view is about the molecule’s high-level description, *e.g.*, biological knowledge graph describes the relations between molecules and other biological entities, and textual description provides annotations for the molecule’s chemical or physical properties.

Molecule Representation with Topology. The most challenging yet fundamental research problem in the foundation model for molecule discovery is molecule representation. The most commonly used data structures are molecule’s topology, *i.e.*, the molecular graph with atoms connected with bonds. As shown in Figure 1, there are three ways to represent a molecule’s topology: (1) Fingerprint is to encode the molecules into a bit vector using hashing method and conducting the message passing along the topology. (2) String representation (*e.g.*, SMILES [259] and SELFIES [127]) is to delegate the molecular graph into a chemical formulation following certain rules. (3) The topological treats the molecule as a graph with atoms and bonds as nodes and edges, respectively. Existing works [44, 140] have empirically

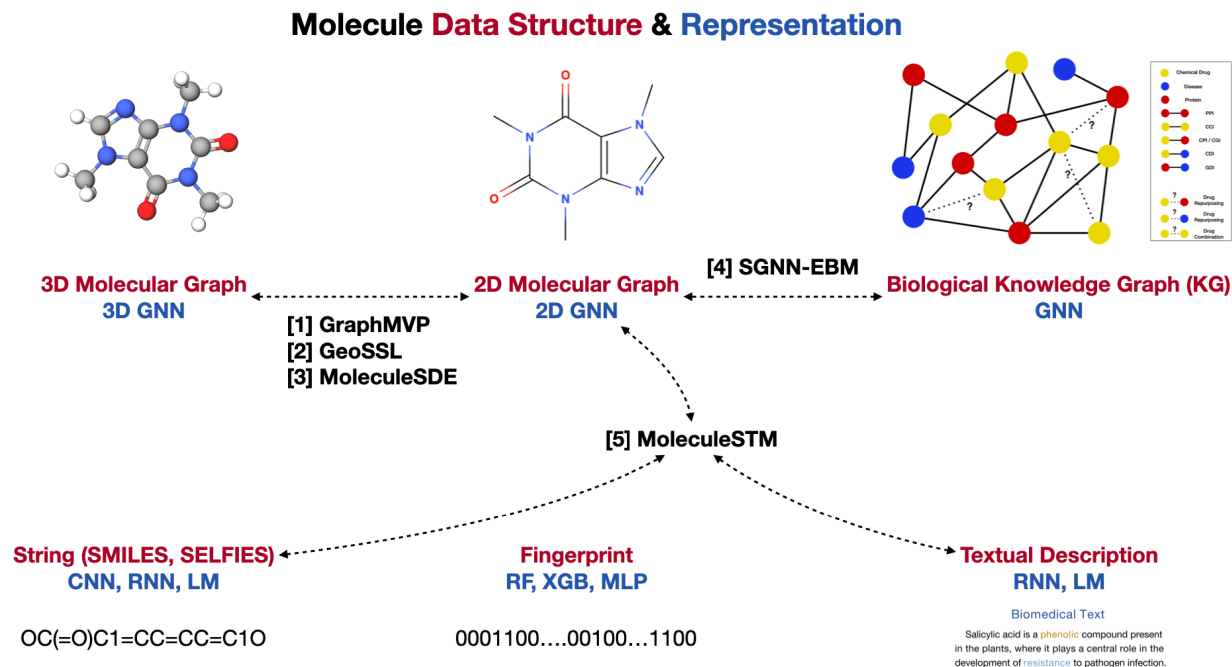


Figure 1. An overview on molecule representation. Five works related to geometry, knowledge graph, and textual description are in **bold** and will be discussed in this report.

shown that the 2D molecular graph is generally superior to the string- and fingerprint-based methods. Meanwhile, molecules are by nature in the form of point clouds in the 3D Euclidean space, and the representation methods based on the 3D geometry can thus lead to more optimal results [144], especially for quantum mechanical tasks and binding tasks.

Molecule Representation with Geometry. In terms of the internal view, molecules can also be naturally represented as 3D point clouds, and the corresponding 3D representation [22, 61, 125, 126, 159, 202, 207, 208, 214] has been widely studied. Along this research line, I am interested in how to adapt unsupervised pretraining or self-supervised learning (SSL) [181, 203] for molecule representation. The main advantage of SSL is that it does not require the expensive data annotation process and instead learns the inherent structure information of the data itself in an unsupervised manner. This is appealing for molecule discovery since over 100M molecules with 2D topologies exist in the public data source [121], yet the chemical properties are largely missing. With this motivation, I have a series of works exploring the effect of geometrical pretraining: (1) GraphMVP [154] and MoleculeSDE [143] conduct molecule topological pretraining with 3D geometries, and (2) GeoSSL [146] explores the molecule pretraining in a pure 3D setting (no covalent bonds). The empirical results reveal that utilizing the 3D geometrical information in the pretraining task can produce a more expressive and more robust molecule representation. Additionally, I propose a novel framework for solving the pretraining task using the energy-based model (EBM) [143, 146, 154], which is flexible for modeling the structured data like molecules.

Molecule Representation with Knowledge Graph. In addition to the internal view, molecules also possess external views. The first external view is the biological knowledge graph (bioKG). Such a knowledge graph can be treated as a high-level description of each molecule’s relation with other biological entities like molecules and diseases. Existing works have explored how to adapt this for protein representation [281] and drug out-of-distribution prediction [271]. However, there is one inherent limitation of the bioKG, which is the data size. This is because a high-quality bioKG should contain most biological entities connected to some extent, while most bioKG can be quite sparse. Thus the applicable setting of using bioKG for molecule representation needs to be carefully considered. Along this direction, I propose a multi-task learning method coined SGNN-EBM [150]. It focuses on the knowledge transfer among tasks specifically for biological assay tasks by explicitly modeling the task distribution.

Molecule Representation with Textual Description. Recently, the foundation model has revolutionized the machine learning community. The key idea is to adopt the natural language as a bridge to fill the gap among different modalities to achieve universal functions. In my most recent work, MoleculeSTM [149], I have started to explore using the large language model to handle challenging molecule tasks, especially in a zero-shot manner. What’s more important, we have shown that the foundation model is indeed a promising direction for solving actual challenging drug discovery tasks.

In the following chapters, as shown in Figure 1, I will illustrate five of my recent works expanding around these two views: GraphMVP, GeoSSL, and MoleculeSDE for molecule internal view, and SGNN-EBM and MoleculeSTM utilizing the molecule’s external view.

First Article.

Pre-training Molecular Graph Representation with 3D Geometry

by

Shengchao Liu^{1,2}, Hanchen Wang³, Weiyang Liu^{3,4},
Joan Lasenby³, Hongyu Guo⁵, and Jian Tang^{1,6,7}

- (¹) Université de Montréal, Montréal, QC, Canada
- (²) Mila-Québec Artificial Intelligence Institute, Montréal, QC, Canada
- (³) University of Cambridge, Cambridge, United Kingdom
- (⁴) MPI for Intelligent Systems, Tübingen, Germany
- (⁵) National Research Council Canada, Ottawa, ON, Canada
- (⁶) HEC Montréal, Montréal, QC, Canada
- (⁷) Canadian Institute for Advanced Research, Toronto, ON, Canada

This article was published in Proceedings of International Conference on Learning Representations (ICLR) 2022 .

The main contributions of Shengchao Liu for this articles are presented as follows:

- Propose the idea;
- Conduct the experiments;
- Write the paper.

Hanchen Wang helped run the ablation studies and paper writing; Weiyang Liu contributed to the paper writing and providing theory analysis; Joan Lasenby, Hongyu Guo, and Jian Tang helped with discussion.

ABSTRACT. Molecular graph representation learning is a fundamental problem in modern drug and material discovery. Molecular graphs are typically modeled by their 2D topological structures, but it has been recently discovered that 3D geometric information plays a more vital role in predicting molecular functionalities. However, the lack of 3D information in real-world scenarios has significantly impeded the learning of geometric graph representation. To cope with this challenge, we propose the Graph Multi-View Pre-training (GraphMVP) framework where self-supervised learning (SSL) is performed by leveraging the correspondence and consistency between 2D topological structures and 3D geometric views. GraphMVP effectively learns a 2D molecular graph encoder that is enhanced by richer and more discriminative 3D geometry. We further provide theoretical insights to justify the effectiveness of GraphMVP. Finally, comprehensive experiments show that GraphMVP can consistently outperform existing graph SSL methods. Code is available on GitHub.

Keywords: Multi-modal pretraining; SSL; topology; geometry; MI; EBM; drug discovery.

1. Introduction

In recent years, drug discovery has drawn increasing interest in the machine learning community. Among many challenges therein, how to discriminatively represent a molecule with a vectorized embedding remains a fundamental yet open challenge. The underlying problem can be decomposed into two components: how to design a common latent space for molecule graphs (*i.e.*, designing a suitable encoder) and how to construct an objective function to supervise the training (*i.e.*, defining a learning target). Falling broadly into the second category, our paper studies self-supervised molecular representation learning by leveraging the consistency between 3D geometry and 2D topology.

Motivated by the prominent success of the pretraining-finetuning pipeline [45], unsupervisedly pre-trained graph neural networks for molecules yields promising performance on downstream tasks and becomes increasingly popular [99, 140, 227, 247, 273, 274]. The key to pre-training lies in finding an effective proxy task (*i.e.*, training objective) to leverage the power of large unlabeled datasets. Inspired by [140, 159, 206] that molecular properties [69, 140] can be better predicted by 3D geometry due to its encoded energy knowledge, we aim to make use of the 3D geometry of molecules in pre-training. However, the stereochemical structures are often very expensive to obtain, making such 3D geometric information scarce in downstream tasks. To address this problem, we propose the Graph Multi-View Pre-training (GraphMVP) framework, where a 2D molecule encoder is pre-trained with the knowledge of 3D geometry and then fine-tuned on downstream tasks without 3D information. Our learning paradigm, during pre-training, injects the knowledge of 3D molecular

geometry to a 2D molecular graph encoder such that the downstream tasks can benefit from the implicit 3D geometric prior even if there is no 3D information available.

We attain the aforementioned goal by leveraging two pretext tasks on the 2D and 3D molecular graphs: one contrastive and one generative SSL. Contrastive SSL creates the supervised signal at an **inter-molecule** level: the 2D and 3D graph pairs are positive if they are from the same molecule, and negative otherwise; Then contrastive SSL [253] will align the positive pairs and contrast the negative pairs simultaneously. Generative SSL [88, 123, 249], on the other hand, obtains the supervised signal in an **intra-molecule** way: it learns a 2D/3D representation that can reconstruct its 3D/2D counterpart view for each molecule itself. To cope with the challenge of measuring the quality of reconstruction on molecule 2D and 3D space, we further propose a novel surrogate objective function called variation representation reconstruction (VRR) for the generative SSL task, which can effectively measure such quality in the continuous representation space. The knowledge acquired by these two SSL tasks is complementary, so our GraphMVP framework integrates them to form a more discriminative 2D molecular graph representation. Consistent and significant performance improvements empirically validate the effectiveness of GraphMVP.

We give additional insights to justify the effectiveness of GraphMVP. First, GraphMVP is a self-supervised learning approach based on maximizing mutual information (MI) between 2D and 3D views, enabling the learnt representation to capture high-level factors [10, 12, 239] in molecule data. Second, we find that 3D molecular geometry is a form of privileged information [244, 245]. It has been proven that using privileged information in training can accelerate the speed of learning. We note that privileged information is only used in training, while it is not available in testing. This perfectly matches our intuition of pre-training molecular representation with 3D geometry.

Our contributions include (1) To our best knowledge, we are the first to incorporate the 3D geometric information into graph SSL; (2) We propose one contrastive and one generative SSL tasks for pre-training. Then we elaborate their difference and empirically validate that combining both can lead to a better representation; (3) We provide theoretical insights and case studies to justify why adding 3D geometry is beneficial; (4) We achieve the SOTA performance among all the SSL methods.

Related work. We briefly review the most related works here and include a more detailed summarization in Appendix A.1. Self-supervised learning (SSL) methods have attracted massive attention to graph applications [158, 160, 262, 265]. In general, there are roughly two categories of graph SSL: contrastive and generative, where they differ on the design of the supervised signals. Contrastive graph SSL [99, 227, 247, 273, 274] constructs the supervised signals at the **inter-graph** level and learns the representation by contrasting with other graphs, while generative graph SSL [82, 99, 101, 140] focuses on reconstructing the original graph at the **intra-graph** level. One of the most significant differences that

separate our work from existing methods is that all previous methods **merely** focus on 2D molecular topology. However, for scientific tasks such as molecular property prediction, 3D geometry should be incorporated as it provides complementary and comprehensive information [159, 206]. To fill this gap, we propose GraphMVP to leverage the 3D geometry in graph self-supervised pre-training.

2. Preliminaries

We first outline the key concepts and notations used in this work. Self-supervised learning (SSL) is based on the *view* design, where each view provides a specific aspect and modality of the data. Each molecule has two natural views: the 2D graph incorporates the topological structure defined by the adjacency, while the 3D graph can better reflect the geometry and spatial relation. From a chemical perspective, 3D geometric graphs focus on the *energy* while 2D graphs emphasize the *topological* information; thus they can be composed for learning more informative representation in GraphMVP. *Transformation* is an atomic operation in SSL that can extract specific information from each view. Next, we will briefly introduce how to represent these two views.

2D Molecular Graph represents molecules as 2D graphs, with atoms as nodes and bonds as edges respectively. We denote each 2D graph as $g_{2D} = (X, E)$, where X is the atom attribute matrix and E is the bond attribute matrix. Notice that here E also includes the bond connectivity. Then we will apply a transformation function T_{2D} on the topological graph. Given a 2D molecular graph g_{2D} , its representation h_{2D} can be obtained from a *2D graph neural network (GNN)* model:

$$h_{2D} = \text{GNN-2D}(T_{2D}(g_{2D})) = \text{GNN-2D}(T_{2D}(X, E)). \quad (2.1)$$

3D Molecular Graph additionally includes spatial positions of the atoms, and they are needless to be static since atoms are in continual motion on *a potential energy surface* [7].

¹ The 3D structures at the local minima on this surface are named *conformer*. As the molecular properties are conformers ensembled [85], GraphMVP provides a novel perspective on adopting 3D conformers for learning better representation. Given a conformer $g_{3D} = (X, R)$, its representation via a *3D GNN* model is:

$$h_{3D} = \text{GNN-3D}(T_{3D}(g_{3D})) = \text{GNN-3D}(T_{3D}(X, R)), \quad (2.2)$$

where R is the 3D-coordinate matrix and T_{3D} is the 3D transformation. In what follows, for notation simplicity, we use \mathbf{x} and \mathbf{y} for the 2D and 3D graphs, *i.e.*, $\mathbf{x} \triangleq g_{2D}$ and $\mathbf{y} \triangleq g_{3D}$. Then the latent representations are denoted as $h_{\mathbf{x}}$ and $h_{\mathbf{y}}$.

¹A more rigorous way of defining conformer is in [173]: a conformer is an isomer of a molecule that differs from another isomer by the rotation of a single bond in the molecule.

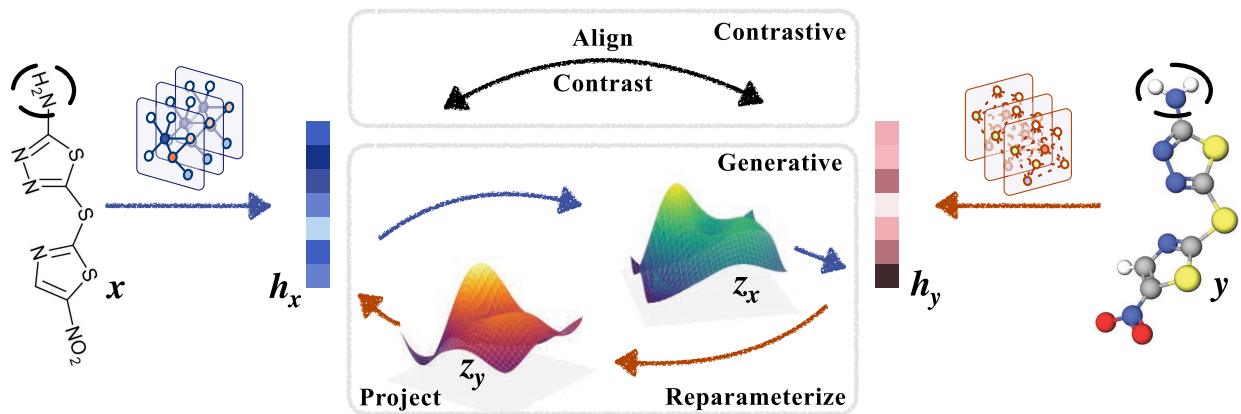


Figure 2. Overview of the pre-training stage in GraphMVP. The black dashed circles denote subgraph masking, and we mask the same region in the 2D and 3D graphs. Multiple views of the molecules (herein: Halicin) are mapped to the representation space via 2D and 3D GNN models, where we conduct GraphMVP for SSL pre-training, using both contrastive and generative pretext tasks.

3. GraphMVP: Graph Multi-View Pre-training

Our model, termed as Graph Multi-View Pre-training (GraphMVP), conducts self-supervised learning (SSL) pre-training with 3D information. The 3D conformers encode rich information about the molecule energy and spatial structure, which are complementary to the 2D topology. Thus, applying SSL between the 2D and 3D views will provide a better 2D representation, which implicitly embeds the ensembles of energies and geometric information for molecules.

In the following, we first present an overview of GraphMVP, and then introduce two pretext tasks specialized concerning 3D conformation structures. Finally, we summarize a broader graph SSL family that prevails the 2D molecular graph representation learning with 3D geometry.

3.1. Overview of GraphMVP

In general, GraphMVP exerts 2D topology and 3D geometry as two complementary views for each molecule. By performing SSL between these views, it is expected to learn a 2D representation enhanced with 3D conformation, which can better reflect certain molecular properties.

As generic SSL pre-training pipelines, GraphMVP has two stages: pre-training then fine-tuning. In the pre-training stage, we conduct SSL via auxiliary tasks on data collections that provide both 2D and 3D molecular structures. During fine-tuning, the pre-trained 2D GNN

models are subsequently fine-tuned on specific downstream tasks, where only 2D molecular graphs are available.

At the SSL pre-training stage, we design two pretext tasks: one contrastive and one generative. We conjecture and then empirically prove that these two tasks are focusing on different learning aspects, which are summarized into the following two points. (1) From the perspective of representation learning, contrastive SSL utilizes **inter-data** knowledge and generative SSL utilizes **intra-data** knowledge. For contrastive SSL, one key step is to obtain the negative view pairs for inter-data contrasting; while generative SSL focuses on each data point itself, by reconstructing the key features at an intra-data level. (2) From the perspective of distribution learning, contrastive SSL and generative SSL are learning the data distribution from a **local** and **global** manner, respectively. Contrastive SSL learns the distribution locally by contrasting the pairwise distance at an inter-data level. Thus, with sufficient number of data points, the local contrastive operation can iteratively recover the data distribution. Generative SSL, on the other hand, learns the global data density function directly.

Therefore, contrastive and generative SSL are essentially conducting representation and distribution learning with different intuitions and disciplines, and we expect that combining both can lead to a better representation. We later carry out an ablation study (Section 4.4) to verify this empirically. In addition, to make the pretext tasks more challenging, we take views for each molecule by randomly masking M nodes (and corresponding edges) as the transformation function, *i.e.*, $T_{2D} = T_{3D} = \text{mask}$. This trick has been widely used in graph SSL [99, 273, 274] and has shown robust improvements.

3.2. Contrastive Self-Supervised Learning between 2D and 3D Views

The main idea of contrastive self-supervised learning (SSL) [28, 181] is first to define positive and negative pairs of views from an inter-data level, and then to align the positive pairs and contrast the negative pairs simultaneously [253]. For each molecule, we first extract representations from 2D and 3D views, *i.e.*, $h_{\mathbf{x}}$ and $h_{\mathbf{y}}$. Then we create positive and negative pairs for contrastive learning: the 2D-3D pairs (\mathbf{x}, \mathbf{y}) for the same molecule are treated as positive, and negative otherwise. Finally, we align the positive pairs and contrast the negative ones. The pipeline is shown in Figure 2. In the following, we discuss two common objective functions on contrastive graph SSL.

InfoNCE is first proposed in [181], and its effectiveness has been validated both empirically [28, 87] and theoretically [5]. Its formulation is given as follows:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{2} \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} \left[\log \frac{\exp(f_{\mathbf{x}}(\mathbf{x}, \mathbf{y}))}{\exp(f_{\mathbf{x}}(\mathbf{x}, \mathbf{y})) + \sum_j \exp(f_{\mathbf{x}}(\mathbf{x}^j, \mathbf{y}))} + \log \frac{\exp(f_{\mathbf{y}}(\mathbf{y}, \mathbf{x}))}{\exp(f_{\mathbf{y}}(\mathbf{y}, \mathbf{x})) + \sum_j \exp(f_{\mathbf{y}}(\mathbf{y}^j, \mathbf{x}))} \right], \quad (3.1)$$

where $\mathbf{x}^j, \mathbf{y}^j$ are randomly sampled 2D and 3D views regarding to the anchored pair (\mathbf{x}, \mathbf{y}) . $f_x(\mathbf{x}, \mathbf{y})$ and $f_y(\mathbf{y}, \mathbf{x})$ are scoring functions for the two corresponding views, with flexible formulations. Here we adopt $f_x(\mathbf{x}, \mathbf{y}) = f_y(\mathbf{y}, \mathbf{x}) = \langle h_x, h_y \rangle$. More details are in Appendix A.4.

Energy-Based Model with Noise Contrastive Estimation (EBM-NCE) is an alternative that has been widely used in the line of graph contrastive SSL [99, 227, 273, 274]. Its intention is essentially the same as InfoNCE, to align positive pairs and contrast negative pairs, while the main difference is the usage of binary cross-entropy and extra noise distribution for negative sampling:

$$\begin{aligned} \mathcal{L}_{\text{EBM-NCE}} = & -\frac{1}{2} \mathbb{E}_{p(\mathbf{y})} \left[\mathbb{E}_{p_n(\mathbf{x}|\mathbf{y})} \log(1 - \sigma(f_x(\mathbf{x}, \mathbf{y}))) + \mathbb{E}_{p(\mathbf{x}|\mathbf{y})} \log \sigma(f_x(\mathbf{x}, \mathbf{y})) \right] \\ & -\frac{1}{2} \mathbb{E}_{p(\mathbf{x})} \left[\mathbb{E}_{p_n(\mathbf{y}|\mathbf{x})} \log(1 - \sigma(f_y(\mathbf{y}, \mathbf{x}))) + \mathbb{E}_{p(\mathbf{y}|\mathbf{x})} \log \sigma(f_y(\mathbf{y}, \mathbf{x})) \right], \end{aligned} \quad (3.2)$$

where p_n is the noise distribution and σ is the sigmoid function. We also notice that the final formulation of EBM-NCE shares certain similarities with Jensen-Shannon estimation (JSE) [180]. However, the derivation process and underlying intuition are different: EBM-NCE models the conditional distributions in MI lower bound (Equation (5.1)) with EBM, while JSE is a special case of variational estimation of f-divergence. Since this is not the main focus of GraphMVP, we expand the a more comprehensive comparison in Appendix A.4, plus the potential benefits with EBM-NCE.

Few works [83] have witnessed the effect on the choice of objectives in graph contrastive SSL. In GraphMVP, we treat it as a hyper-parameter and further run ablation studies on them, *i.e.*, to solely use either InfoNCE ($\mathcal{L}_C = \mathcal{L}_{\text{InfoNCE}}$) or EMB-NCE ($\mathcal{L}_C = \mathcal{L}_{\text{EBM-NCE}}$).

3.3. Generative Self-Supervised Learning between 2D and 3D Views

Generative SSL is another classic track for unsupervised pre-training [29, 122, 123, 131]. It aims at learning an effective representation by self-reconstructing each data point. Specifically to drug discovery, we have one 2D graph and a certain number of 3D conformers for each molecule, and our goal is to learn a robust 2D/3D representation that can, to the most extent, recover its 3D/2D counterparts. By doing so, generative SSL can enforce 2D/3D GNN to encode the most crucial geometry/topology information, which can improve the downstream performance.

There are many options for generative models, including variational auto-encoder (VAE) [123], generative adversarial networks (GAN) [73], flow-based model [47], etc. In GraphMVP, we prefer VAE-like method for the following reasons: (1) The mapping between two molecular views is stochastic: multiple 3D conformers correspond to the same 2D topology; (2) An explicit 2D graph representation (*i.e.*, feature encoder) is required for

downstream tasks; (3) Decoders for structured data such as graph are often highly nontrivial to design, which make them a suboptimal choice.

Variational Molecule Reconstruction. Therefore we propose a *light* VAE-like generative SSL, equipped with a *crafty* surrogate loss, which we describe in the following. We start with an example for illustration. When generating 3D conformers from their corresponding 2D topology, we want to model the conditional likelihood $p(\mathbf{y}|\mathbf{x})$. By introducing a reparameterized variable $\mathbf{z}_x = \mu_x + \sigma_x \odot \epsilon$, where μ_x and σ_x are two flexible functions on h_x , $\epsilon \sim \mathcal{N}(0, I)$ and \odot is the element-wise production, we have the following lower bound:

$$\log p(\mathbf{y}|\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}_x|\mathbf{x})}[\log p(\mathbf{y}|\mathbf{z}_x)] - KL(q(\mathbf{z}_x|\mathbf{x})||p(\mathbf{z}_x)). \quad (3.3)$$

The expression for $\log p(\mathbf{x}|\mathbf{y})$ can be similarly derived. Equation (3.3) includes a conditional log-likelihood and a KL-divergence term, where the bottleneck is to calculate the first term for structured data. This term has also been recognized as the *reconstruction term*: it is essentially to reconstruct the 3D conformers (\mathbf{y}) from the sampled 2D molecular graph representation (\mathbf{z}_x). However, performing the graph reconstruction on the data space is not trivial: since molecules (*e.g.*, atoms and bonds) are discrete, modeling and measuring on the molecule space will bring extra obstacles.

Variational Representation Reconstruction (VRR). To cope with this challenge, we propose a novel surrogate loss by switching the reconstruction from data space to representation space. Instead of decoding the latent code z_x to data space, we can directly project it to the 3D representation space, denoted as $q_x(z_x)$. Since the representation space is continuous, we may as well model the conditional log-likelihood with Gaussian distribution, resulting in L2 distance for reconstruction, *i.e.*, $\|q_x(z_x) - \text{SG}(h_y(\mathbf{y}))\|^2$. Here SG is the stop-gradient operation, assuming that h_y is a fixed learnt representation function. SG has been widely adopted in the SSL literature to avoid model collapse [30, 75]. We call this surrogate loss as variational representation reconstruction (VRR):

$$\begin{aligned} \mathcal{L}_G = \mathcal{L}_{\text{VRR}} = & \frac{1}{2} \left[\mathbb{E}_{q(\mathbf{z}_x|\mathbf{x})} [\|q_x(\mathbf{z}_x) - \text{SG}(h_y)\|^2] + \mathbb{E}_{q(\mathbf{z}_y|\mathbf{y})} [\|q_y(\mathbf{z}_y) - \text{SG}(h_x)\|_2^2] \right] \\ & + \frac{\beta}{2} \cdot \left[KL(q(\mathbf{z}_x|\mathbf{x})||p(\mathbf{z}_x)) + KL(q(\mathbf{z}_y|\mathbf{y})||p(\mathbf{z}_y)) \right]. \end{aligned} \quad (3.4)$$

We give a simplified illustration for the generative SSL pipeline in Figure 2 and the complete derivations in Appendix A.5. As will be discussed in Section 5.1, VRR is actually maximizing MI, and MI is invariant to continuous bijective function [12]. Thus, this surrogate loss would be exact if the encoding function h satisfies this condition. However, we find that GNN, though does not meet the condition, can provide quite robust performance, which empirically justify the effectiveness of VRR.

3.4. Multi-task Objective Function

As discussed before, contrastive SSL and generative SSL essentially learn the representation from distinct viewpoints. A reasonable conjecture is that combining both SSL methods can lead to overall better performance, thus we arrive at minimizing the following complete objective for GraphMVP:

$$\mathcal{L}_{\text{GraphMVP}} = \alpha_1 \cdot \mathcal{L}_C + \alpha_2 \cdot \mathcal{L}_G, \quad (3.5)$$

where α_1, α_2 are weighting coefficients. A later performed ablation study (Section 4.4) delivers two important messages: (1) Both individual contrastive and generative SSL on 3D conformers can consistently help improve the 2D representation learning; (2) Combining the two SSL strategies can yield further improvements. Thus, we draw the conclusion that GraphMVP (Equation (3.5)) is able to obtain an augmented 2D representation by fully utilizing the 3D information.

As discussed in Section 1, existing graph SSL methods only focus on the 2D topology, which is in parallel to GraphMVP: 2D graph SSL focuses on exploiting the 2D structure topology, and GraphMVP takes advantage of the 3D geometry information. Thus, we propose to merge the 2D SSL into GraphMVP. Since there are two main categories in 2D graph SSL: generative and contrastive, we propose two variants GraphMVP-G and GraphMVP-C accordingly. Their objectives are as follows:

$$\begin{aligned} \mathcal{L}_{\text{GraphMVP-G}} &= \mathcal{L}_{\text{GraphMVP}} + \alpha_3 \cdot \mathcal{L}_{\text{Generative 2D-SSL}}, \\ \mathcal{L}_{\text{GraphMVP-C}} &= \mathcal{L}_{\text{GraphMVP}} + \alpha_3 \cdot \mathcal{L}_{\text{Contrastive 2D-SSL}}. \end{aligned} \quad (3.6)$$

Later, the empirical results also help support the effectiveness of GraphMVP-G and GraphMVP-C, and thus, we can conclude that existing 2D SSL is complementary to GraphMVP.

4. Experiments and Results

4.1. Experimental Settings

Datasets. We pre-train models on the same dataset then fine-tune on the wide range of downstream tasks. We randomly select 50k qualified molecules from GEOM [7] with both 2D and 3D structures for the pre-training. As clarified in Section 3.1, conformer ensembles can better reflect the molecular property, thus we take C conformers of each molecule. For downstream tasks, we first stick to the same setting of the main graph SSL work [99, 273, 274], exploring 8 binary molecular property prediction tasks, which are all in the low-data regime. Then we explore 6 regression tasks from various low-data domains to be more comprehensive. We describe all the datasets in Appendix A.6.

2D GNN. We follow the research line of SSL on molecule graph [99, 273, 274], using the same Graph Isomorphism Network (GIN) [266] as the backbone model, with the same feature sets.

3D GNN. We choose SchNet [206] for geometric modeling, since SchNet: (1) is found to be a strong geometric representation learning method under the fair benchmarking; (2) can be trained more efficiently, comparing to the other recent 3D models. More detailed explanations are in Appendix A.2.2.

4.2. Main Results on Molecular Property Prediction.

We carry out comprehensive comparisons with 10 SSL baselines and random initialization. For pre-training, we apply all SSL methods on the same dataset based on GEOM [7]. For fine-tuning, we follow the same setting [99, 273, 274] with 8 low-data molecular property prediction tasks.

Baselines. Due to the rapid growth of graph SSL [160, 262, 265], we are only able to benchmark the most well-acknowledged baselines: EdgePred [82], InfoGraph [227], GPT-GNN[101], AttrMask & ContextPred[99], GraphLoG[267], G- $\{\text{Contextual, Motif}\}$ [200], GraphCL[274], JOAO[273].

Our method. GraphMVP has two key factors: i) masking ratio (M) and ii) number of conformers for each molecule (C). We set $M = 0.15$ and $C = 5$ by default, and will explore their effects in the following ablation studies in Section 4.3. For EBM-NCE loss, we adopt the empirical distribution for noise distribution. For Equation (3.6), we pick the empirically optimal generative and contrastive 2D SSL method: that is AttrMask for GraphMVP-G and ContextPred for GraphMVP-C.

The main results on 8 molecular property prediction tasks are listed in Table 1. We observe that the performance of GraphMVP is significantly better than the random initialized one, and the average performance outperforms the existing SSL methods by a large margin. In addition, GraphMVP-G and GraphMVP-C consistently improve the performance, supporting the claim: **3D geometry is complementary to the 2D topology**. GraphMVP leverages the information between 3D geometry and 2D topology, and 2D SSL plays the role as regularizer to extract more 2D topological information; they are extracting different perspectives of information and are indeed complementary to each other.

4.3. Ablation Study: The Effect of Masking Ratio and Number of Conformers

We analyze the effects of masking ratio M and the number of conformers C in GraphMVP. In Table 1, we set the M as 0.15 since it has been widely used in existing SSL methods [99, 273, 274], and C is set to 5, which we will explain below. We explore on the range of

Table 1. Results for molecular property prediction tasks. For each downstream task, we report the mean (and standard deviation) ROC-AUC of 3 seeds with scaffold splitting. For GraphMVP, we set $M = 0.15$ and $C = 5$. The best and second best results are marked **bold** and **bold**, respectively.

Pre-training	BBBP	Tox21	ToxCast	Sider	ClinTox	MUV	HIV	Bace	Avg
–	65.4(2.4)	74.9(0.8)	61.6(1.2)	58.0(2.4)	58.8(5.5)	71.0(2.5)	75.3(0.5)	72.6(4.9)	67.21
EdgePred	64.5(3.1)	74.5(0.4)	60.8(0.5)	56.7(0.1)	55.8(6.2)	73.3(1.6)	75.1(0.8)	64.6(4.7)	65.64
AttrMask	70.2(0.5)	74.2(0.8)	62.5(0.4)	60.4(0.6)	68.6(9.6)	73.9(1.3)	74.3(1.3)	77.2(1.4)	70.16
GPT-GNN	64.5(1.1)	75.3(0.5)	62.2(0.1)	57.5(4.2)	57.8(3.1)	76.1(2.3)	75.1(0.2)	77.6(0.5)	68.27
InfoGraph	69.2(0.8)	73.0(0.7)	62.0(0.3)	59.2(0.2)	75.1(5.0)	74.0(1.5)	74.5(1.8)	73.9(2.5)	70.10
ContextPred	71.2(0.9)	73.3(0.5)	62.8(0.3)	59.3(1.4)	73.7(4.0)	72.5(2.2)	75.8(1.1)	78.6(1.4)	70.89
GraphLoG	67.8(1.7)	73.0(0.3)	62.2(0.4)	57.4(2.3)	62.0(1.8)	73.1(1.7)	73.4(0.6)	78.8(0.7)	68.47
G-Contextual	70.3(1.6)	75.2(0.3)	62.6(0.3)	58.4(0.6)	59.9(8.2)	72.3(0.9)	75.9(0.9)	79.2(0.3)	69.21
G-Motif	66.4(3.4)	73.2(0.8)	62.6(0.5)	60.6(1.1)	77.8(2.0)	73.3(2.0)	73.8(1.4)	73.4(4.0)	70.14
GraphCL	67.5(3.3)	75.0(0.3)	62.8(0.2)	60.1(1.3)	78.9(4.2)	77.1(1.0)	75.0(0.4)	68.7(7.8)	70.64
JOAO	66.0(0.6)	74.4(0.7)	62.7(0.6)	60.7(1.0)	66.3(3.9)	77.0(2.2)	76.6(0.5)	72.9(2.0)	69.57
GraphMVP	68.5(0.2)	74.5(0.4)	62.7(0.1)	62.3(1.6)	79.0(2.5)	75.0(1.4)	74.8(1.4)	76.8(1.1)	71.69
GraphMVP-G	70.8(0.5)	75.9(0.5)	63.1(0.2)	60.2(1.1)	79.1(2.8)	77.7(0.6)	76.0(0.1)	79.3(1.5)	72.76
GraphMVP-C	72.4(1.6)	74.4(0.2)	63.1(0.4)	63.9(1.2)	77.5(4.2)	75.0(1.0)	77.0(1.2)	81.2(0.9)	73.07

Table 2. Ablation of masking ratio M , $C \equiv 5$.

M	GraphMVP	GraphMVP-G	GraphMVP-C
0	71.12	72.15	72.66
0.15	71.60	72.76	73.08
0.30	71.79	72.91	73.17

Table 3. Ablation of # conformer C , $M \equiv 0.15$.

C	GraphMVP	GraphMVP-G	GraphMVP-C
1	71.61	72.80	72.46
5	71.60	72.76	73.08
10	72.20	72.59	73.09
20	72.39	73.00	73.02

$M \in \{0, 0.15, 0.3\}$ and $C \in \{1, 5, 10, 20\}$, and report the average performance. The complete results are in Appendix A.7.2.

As seen in Table 2, the improvement is more obvious from $M = 0$ (raw graph) to $M = 0.15$ than from $M = 0.15$ to $M = 0.3$. This can be explained that subgraph masking with larger ratio will make the SSL tasks more challenging, especially comparing to the raw graph ($M = 0$).

Table 3 shows the effect for C . We observe that the performance is generally better when adding more conformers, but will reach a plateau above certain thresholds. This observation matches with previous findings [8]: adding more conformers to augment the representation learning is not as helpful as expected; while we conclude that adding more conformers can be beneficial with little improvement. One possible reason is, when generating the dataset, we are sampling top- C conformers with highest possibility and lowest energy. In other words, top-5 conformers are sufficient to cover the most conformers with equilibrium state (over 80%), and the effect of larger C is thus modest.

To sum up, adding more conformers might be helpful, but the computation cost can grow linearly with the increase in dataset size. On the other hand, enlarging the masking ratio will

Table 4. Ablation on the objective function.

GraphMVP Loss	Contrastive	Generative	Avg
Random			67.21
InfoNCE only	✓		68.85
EBM-NCE only	✓		70.15
VRR only		✓	69.29
RR only		✓	68.89
InfoNCE + VRR	✓	✓	70.67
EBM-NCE + VRR	✓	✓	71.69
InfoNCE + RR	✓	✓	70.60
EBM-NCE + RR	✓	✓	70.94

not induce extra cost, yet the performance is slightly better. Therefore, we would encourage tuning masking ratios prior to trying a larger number of conformers from the perspective of efficiency and effectiveness.

4.4. Ablation Study: The Effect of Objective Function

In Section 3, we introduce a new contrastive learning objective family called EBM-NCE, and we take either InfoNCE and EBM-NCE as the contrastive SSL. For the generative SSL task, we propose a novel objective function called variational representation reconstruction (VRR) in Equation (3.4). As discussed in Section 3.3, stochasticity is important for GraphMVP since it can capture the conformer distribution for each 2D molecular graph. To verify this, we add an ablation study on *representation reconstruction (RR)* by removing stochasticity in VRR. Thus, here we deploy a comprehensive ablation study to explore the effect for each individual objective function (InfoNCE, EBM-NCE, VRR and RR), followed by the pairwise combinations between them.

The results in Table 4 give certain constructive insights as follows: (1) Each individual SSL objective function (middle block) can lead to better performance. This strengthens the claim that adding 3D information is helpful for 2D representation learning. (2) According to the combination of those SSL objective functions (bottom block), adding both contrastive and generative SSL can consistently improve the performance. This verifies our claim that conducting SSL at both the inter-data and intra-data level is beneficial. (3) We can see VRR is consistently better than RR on all settings, which verifies that stochasticity is an important factor in modeling 3D conformers for molecules.

Table 5. Results for four molecular property prediction tasks (regression) and two DTA tasks (regression). We report the mean RMSE of 3 seeds with scaffold splitting for molecular property downstream tasks, and mean MSE for 3 seeds with random splitting on DTA tasks. For GraphMVP, we set $M = 0.15$ and $C = 5$. The best performance for each task is marked in **bold**. We omit the std here since they are very small and indistinguishable. For complete results, please check Appendix A.7.4.

Pre-training	Molecular Property Prediction					Drug-Target Affinity		
	ESOL	Lipo	Malaria	CEP	Avg	Davis	KIBA	Avg
–	1.178	0.744	1.127	1.254	1.0756	0.286	0.206	0.2459
AM	1.112	0.730	1.119	1.256	1.0542	0.291	0.203	0.2476
CP	1.196	0.702	1.101	1.243	1.0606	0.279	0.198	0.2382
JOAO	1.120	0.708	1.145	1.293	1.0663	0.281	0.196	0.2387
GraphMVP	1.091	0.718	1.114	1.236	1.0397	0.280	0.178	0.2286
GraphMVP-G	1.064	0.691	1.106	1.228	1.0221	0.274	0.175	0.2248
GraphMVP-C	1.029	0.681	1.097	1.244	1.0128	0.276	0.168	0.2223

4.5. Broader Range of Downstream Tasks

The 8 binary downstream tasks discussed so far have been widely applied in the graph SSL research line on molecules [99, 273, 274], but there are more tasks where the 3D conformers can be helpful. Here we test 4 extra regression property prediction tasks and 2 drug-target affinity tasks.

About the dataset statistics, more detailed information can be found in Appendix A.6, and we may as well briefly describe the affinity task here. Drug-target affinity (DTA) is a crucial task [182, 183, 260] in drug discovery, where it models both the molecular drugs and target proteins, with the goal to predict their affinity scores. One recent work [176] is modeling the molecular drugs with 2D GNN and target protein (as an amino-acid sequence) with convolution neural network (CNN). We adopt this setting by pre-training the 2D GNN using GraphMVP. As illustrated in Table 5, the consistent performance gain verifies the effectiveness of our proposed GraphMVP.

4.6. Case Study

We investigate how GraphMVP helps when the task objectives are challenging with respect to the 2D topology but straightforward using 3D geometry (as shown in Figure 3). We therefore design two case studies to testify how GraphMVP transfers knowledge from 3D geometry into the 2D representation.

The first case study is *3D Diameter Prediction*. For molecules, usually, the longer the 2D diameter is, the larger the 3D diameter (largest atomic pairwise l2 distance). However, this does not always hold, and we are interested in using the 2D graph to predict the

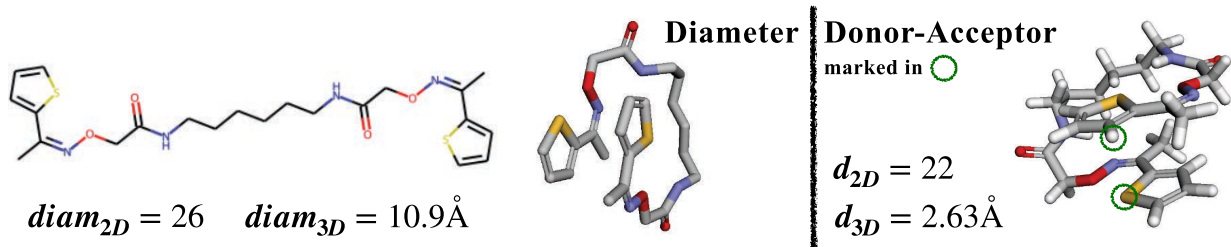


Figure 3. We select the molecules whose properties can be easily resolved via 3D but not 2D. The randomly initialised 2D GNN achieves accuracy of 38.9 ± 0.8 and 77.9 ± 1.1 , respectively. The GraphMVP pre-trained ones obtain scores of 42.3 ± 1.3 and 81.5 ± 0.4 , outperforming all the precedents in Section 4.2. We plot cases where random initialization fails but GraphMVP is correct.

3D diameter. The second case study is *Long-Range Donor-Acceptor Detection*. Molecules possess a special geometric structure called donor-acceptor bond, and we want to use 2D molecular graph to detect this special structure. We validate that GraphMVP consistently brings improvements on these 2 case studies, and provide more detailed discussions and interpretations in Appendix A.7.6.

5. Theoretical Insights

In this section, we provide the mathematical insights behind GraphMVP. We will first discuss both contrastive and generative SSL methods (Sections 3.2 and 3.3) are maximizing the mutual information (MI) and then how the 3D geometry, as privileged information, can help 2D representation learning.

5.1. Maximizing Mutual Information

Mutual information (MI) measures the non-linear dependence [12] between two random variables: the larger MI, the stronger dependence between the variables. Therefore for GraphMVP we can interpret it as maximizing MI between 2D and 3D views: to obtain a more robust 2D/3D representation by sharing more information with its 3D/2D counterparts. This is also consistent with the sample complexity theory [5, 54, 64] where SSL as functional regularizer can reduce the uncertainty in representation learning. We first derive a lower bound for MI (see derivations in Appendix A.3), and the corresponding objective function \mathcal{L}_{MI} is

$$I(X; Y) \geq \mathcal{L}_{\text{MI}} = \frac{1}{2} \mathbb{E}_{p(x,y)} [\log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x}|\mathbf{y})]. \quad (5.1)$$

Contrastive Self-Supervised Learning. InfoNCE was initialized proposed to maximize the MI directly [181]. Here in GraphMVP EBM-NCE estimates the conditional likelihood in Equation (5.1) using EBM, and solves it with NCE [79]. As a result, EBM-NCE can also be seen as maximizing MI between 2D and 3D views. The detailed derivations can be found in Appendix A.4.2.

Generative Self-Supervised Learning. One alternative solution is to use a variational lower bound to approximate the conditional log-likelihood terms in Equation (5.1). Then we can follow the same pipeline in Section 3.3, ending up with the surrogate objective, *i.e.*, VRR in Equation (3.4).

5.2. 3D Geometry as Privileged Information

We show the theoretical insights from privileged information that motivate GraphMVP. We start by considering a supervised learning setting where $(\mathbf{u}_i, \mathbf{l}_i)$ is a feature-label pair and \mathbf{u}_i^* is the privileged information [244, 245]. The privileged information is defined to be additional information about the input $(\mathbf{u}_i, \mathbf{l}_i)$ in order to support the prediction. For example, \mathbf{u}_i could be some CT images of a particular disease, \mathbf{l}_i could be the label of the disease and \mathbf{u}_i^* is the medical report from a doctor. VC theory [243, 244] characterizes the learning speed of an algorithm from the capacity of the algorithm and the amount of training data. Considering a binary classifier f from a function class \mathcal{F} with finite VC-dimension $\text{VCD}(\mathcal{F})$. With probability $1 - \delta$, the expected error is upper bounded by

$$R(f) \leq R_n(f) + \mathcal{O}\left(\left(\frac{\text{VCD}(\mathcal{F}) - \log \delta}{n}\right)^\beta\right) \quad (5.2)$$

where $R_n(f)$ denotes the training error and n is the number of training samples. When the training data is separable, then $R_n(f)$ will diminish to zero and β is equal to 1. When the training data is non-separable, β is $\frac{1}{2}$. Therefore, the rate of convergence for the separable case is of order $1/n$. In contrast, the rate for the non-separable case is of order $1/\sqrt{n}$. We note that such a difference is huge, since the same order of bounds require up to 100 training samples versus 10,000 samples. Privileged information makes the training data separable such that the learning can be more efficient. Connecting the results to GraphMVP, we notice that the 3D geometric information of molecules can be viewed as a form of privileged information, since 3D information can effectively make molecules more separable for some properties [140, 159, 206]. Besides, privileged information is only used in training, and it well matches our usage of 3D geometry for pre-training. In fact, using 3D structures as privileged information has been already shown quite useful in protein classification [245], which serves as a strong evidence to justify the effectiveness of 3D information in graph SSL pre-training.

6. Conclusion and Future Work

In this work, we provide a very general framework, coined GraphMVP. From the domain perspective, GraphMVP (1) is the first to incorporate 3D information for augmenting 2D graph representation learning and (2) is able to take advantages of 3D conformers by considering stochasticity in modeling. From the aspect of technical novelties, GraphMVP brings following insights when introducing 2 SSL tasks: (1) Following Equation (5.1), GraphMVP

proposes EBM-NCE and VRR, where they are modeling the conditional distributions using EBM and variational distribution respectively. (2) EBM-NCE is similar to JSE, while we start with a different direction for theoretical intuition, yet EBM opens another promising venue in this area. (3) VRR, as a generative SSL method, is able to alleviate the potential issues in molecule generation [63, 283]. (4) Ultimately, GraphMVP combines both contrastive SSL (InfoNCE or EBM-NCE) and generative SSL (VRR) for objective function. Both empirical results (solid performance improvements on 14 downstream datasets) and theoretical analysis can strongly support the above domain and technical contributions.

We want to emphasize that GraphMVP is model-agnostic and has the potential to be expanded to many other low-data applications. This motivates broad directions for future exploration, including but not limited to: (1) More powerful 2D and 3D molecule representation methods. (2) Different application domain other than small molecules, *e.g.*, large molecules like proteins.

Second Article.

Molecular Geometry Pretraining with SE(3)-Invariant Denoising Distance Matching

by

Shengchao Liu^{1,2}, Hongyu Guo³, and Jian Tang^{1,4,5}

- (¹) Université de Montréal, Montréal, QC, Canada
- (²) Mila-Québec Artificial Intelligence Institute, Montréal, QC, Canada
- (³) National Research Council Canada, Ottawa, ON, Canada
- (⁴) HEC Montréal, Montréal, QC, Canada
- (⁵) Canadian Institute for Advanced Research, Toronto, ON, Canada

This article was published in Proceedings of International Conference on Learning Representations (ICLR) 2023 .

The main contributions of Shengchao Liu for this articles are presented as follows:

- Propose the idea;
- Conduct the experiments;
- Write the paper.

Hongyu Guo and Jian Tang helped discussion and paper writing.

ABSTRACT. Molecular representation pretraining is critical in various applications for drug and material discovery due to the limited number of labeled molecules, and most existing work focuses on pretraining on 2D molecular graphs. However, the power of pretraining on 3D geometric structures has been less explored. This is owing to the difficulty of finding a sufficient proxy task that can empower the pretraining to effectively extract essential features from the geometric structures. Motivated by the dynamic nature of 3D molecules, where the continuous motion of a molecule in the 3D Euclidean space forms a smooth potential energy surface, we propose GeoSSL, a 3D coordinate denoising pretraining framework to model such an energy landscape. Further by leveraging an SE(3)-invariant score matching method, we propose GeoSSL-DDM in which the coordinate denoising proxy task is effectively boiled down to denoising the pairwise atomic distances in a molecule. Our comprehensive experiments confirm the effectiveness and robustness of our proposed method.

Keywords: Geometry; score matching; SE(3)-invariant; drug discovery.

1. Introduction

Learning effective molecular representations is critical in a variety of tasks in drug and material discovery, such as molecular property prediction [52, 69, 70, 270], *de novo* molecular design and optimization [23, 148, 152, 155, 211, 279], and retrosynthesis and reaction planning [16, 74, 210, 229]. Recent work based on graph neural networks (GNNs) [69] has shown superior performance thanks to the simplicity and effectiveness of GNNs in modeling graph-structured data. However, the problem remains challenging due to the limited number of labeled molecules as it is in general expensive and time-consuming to label molecules, which usually requires expensive physics simulations or wet-lab experiments.

As a result, recently, there has been growing interest in developing pretraining or self-supervised learning methods for learning molecular representations by leveraging the huge amount of unlabeled molecule data [99, 142, 227, 274]. These methods have shown superior performance on many tasks, especially when the number of labeled molecules is insufficient. However, one limitation of these approaches is that they represent molecules as topological graphs, and molecular representations are learned through pretraining 2D topological structures (*i.e.*, based on the covalent bonds). But intrinsically, for molecules, a more natural representation is based on their 3D geometric structures, which largely determine the corresponding physical and chemical properties. Indeed, recent works [69, 154] have empirically verified the importance of applying 3D geometric information for molecular property prediction tasks. Therefore, a more promising direction is to pretrain molecular representations based on their 3D geometric structures, which is the main focus of this paper.

The main challenge for molecule geometric pretraining arises from discovering an effective proxy task to empower the pretraining to extract essential features from the 3D geometric structures. Our proxy task is motivated by the following observations. Studies [204] have shown that molecules are not static but in a continuous motion in the 3D Euclidean space,

forming a potential energy surface (PES). As shown in Figure 4, it is desirable to study the molecule in the local minima of the PES, called *conformer*. However, such stable state conformer often comes with different noises for the following reasons. First, the statistical and systematic errors in conformation estimation are unavoidable [33]. Second, it has been well-acknowledged that a conformer can have vibrations around the local minima in PES. Such characteristics of the molecular geometry motivate us to attempt to denoise the molecular coordinates around the local minima, to mimic the computation errors and conformation vibration within the corresponding local region. The denoising goal is to learn molecular representations that are robust to such noises and effectively capture the energy surface around the local minima.

To achieve the aforementioned goal, we first introduce a general *geometric self-supervised learning* framework called GeoSSL. Based on this, we further propose an *SE(3)-invariant denoising distance matching* pretraining algorithm, GeoSSL-DDM. In a nutshell, to capture the smooth energy surface around the local minima, we aim to maximize the mutual information (MI) between a given *stable geometry* and its *perturbed version* (*i.e.*,

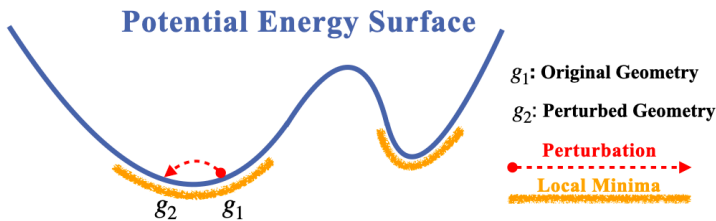


Figure 4. Illustration on coordinate geometry of molecules. The molecule is in a continuous motion, forming a potential energy surface (PES), where each 3D coordinate (x-axis) corresponds to an energy value (y-axis). The provided molecules, *i.e.*, conformers, are in the local minima (g_1). It often comes with noises around the minima (*e.g.*, statistical and systematic errors or vibrations), which can be captured using the perturbed geometry (g_2). In practice, it is difficult to directly maximize the mutual information between two random variables. Thus, we propose to maximize an equivalent lower bound of the above mutual information, which amounts to a pretraining framework on denoising a geometric structure, coined GeoSSL. Moreover, directly denoising such noisy coordinates remains challenging because one may need to effectively constrain the pairwise atomic distances while changing the atomic coordinates. To cope with this obstacle, we further leverage an SE(3)-invariant score matching method, GeoSSL-DDM, to successfully transform the coordinate denoising desire to the denoising of pairwise atomic distances, which then can be effectively computed. In other words, our pretraining proxy task, namely mutual information maximization, effectively boils down to achieving an intuitive learning objective: denoising a molecule’s pairwise atomic distances. Using 22 downstream geometric molecular prediction tasks, we empirically verify that our method outperforms nine pretraining baselines.

Our main contributions are summarized as follows. (1) We propose a novel geometric self-supervised learning framework, GeoSSL. To the best of our knowledge, it is the first

pretraining framework focusing on the pure 3D molecular data ². (2) To overcome the challenge of attaining the coordinate denoising objective in GeoSSL, we propose GeoSSL-DDM, an SE(3)-invariant score matching strategy to successfully transform such objective into the denoising of pairwise atomic distances. (3) We empirically demonstrate the effectiveness and robustness of GeoSSL-DDM on 22 downstream tasks.

2. Related Work

2.1. Equivariant Geometric Molecule Representation Learning

Geometric representation learning. Recently, 3D geometric representation learning has been widely explored in the machine learning community, including but not limited to 3D point clouds [27, 187, 212, 241], N-body particle [189, 202], and 3D molecular conformation [22, 125, 126, 159, 207, 208, 214], amongst many others. The learned representation should satisfy the physical constraints, *e.g.*, it should be equivariant to the rotation and translation in the 3D Euclidean space. Such constraints can be described using group symmetry as introduced below.

SE(3)-invariant energy. Constrained by the physical nature of 3D geometric data, a key principle we need to follow is to learn an SE(3)-equivariant representation function. The SE(3) is the special Euclidean group consisting of rigid transformations in the 3D Cartesian space, where the transformations include all the combinations of translations and rotations. Namely, the learned representation should be equivariant to translations and rotations for molecule geometries. We also note that the representation function needlessly satisfies the reflection equivariance for certain tasks like molecular chirality [6]. For more rigorous discussion, please check [61, 67, 234]. In this work, we will design an SE(3)-invariant energy (score) function in addition to the SE(3)-equivariant representation backbone model.

2.2. Self-Supervised Learning for Molecule Representation Learning

In general, there are two categories of self-supervised learning (SSL) [158, 160, 262, 265]: contrastive and generative, and the main difference is if the supervised signals are constructed in an inter-data or intra-data manner. Contrastive SSL extracts two views from the data and determines the supervised signals by detecting whether the sampled view pairs are from the same data. Generative SSL learns structural information by reconstructing partial information from the data itself.

²During the rebuttal of our submission, one of the reviewers pointed us to this parallel work [278], which is also under review. We provide a detailed comparison with this work in Appendix B.7.

2D molecular graph (topology) self-supervised learning. One of the mainstream research lines for molecule pretraining is on the 2D molecular graph. It treats the molecules as 2D graphs, where atoms and bonds are nodes and edges, respectively. It then carries out a pretraining task by either detecting if the two augmentations (*e.g.*, neighborhood extraction, node dropping, edge dropping, etc) correspond to the same molecular graph [99, 227, 274] or if the representation can successfully reconstruct certain substructures of the molecular graphs [99, 101, 142].

3D molecular graph (geometry) self-supervised learning. Self-supervised learning for 3D molecular graphs is still underexplored. The only related works are [58, 154], which leverage both 2D topology and 3D conformation to improve the molecule representation learning. For example, ChemRL-GEM [58] designs a novel model using 2D and 3D molecular graphs. Regarding SSL, it utilizes the geometry information by conducting distance and angle prediction as the generative pretraining tasks. GraphMVP [154] introduces an extra 2D topology and employs detection and reconstruction tasks simultaneously between 2D and 3D graphs, yet it focuses on 2D downstream tasks due to the small scale of the pretraining dataset. To the best of our knowledge, our work is the first to explicitly do SSL on pure 3D geometry along the molecule representation learning research line. We note that there is a parallel work [278], which is also under review; Appendix B.5 provides a detailed comparison, highlighting the fact that the parallel work is a special case of GeoSSL-DDM.

3. Preliminaries

Molecular geometry graph. Molecules can be naturally featured in a geometric formulation, *i.e.*, all the atoms are spatially located in 3D Euclidean space. Note that the covalent bonds are added heuristically by expert rules, so they are only applicable in 2D topology graphs. Besides, atoms are not static but in a continual motion along a potential energy surface [7]. The 3D structures at the local minima on this surface are named *conformer*, as shown in Figure 4. Conformers at such an equilibrium state possess nice properties, and we would like to model them during pretraining.

Geometric neural network. We denote each conformer as $\mathbf{g} = (X, R)$. Here $X \in \mathbb{R}^{n \times d}$ is the atom attribute matrix and $R \in \mathbb{R}^{n \times 3}$ is the atom 3D-coordinate matrix, where n is the number of atoms and d is the feature dimension. The representations for the i -th node and whole molecule are:

$$h_i = \text{GNN-3D}(T(\mathbf{g}))_i = \text{GNN-3D}(T(X, R))_i, \quad h = \text{READOUT}(h_0, \dots, h_{n-1}), \quad (3.1)$$

where T is the transformation function like atom masking, and READOUT is the readout function. In this work, we take the mean over all the node representations as the readout function.

Energy-based model and denoising score matching. Energy-based model (EBM) is a flexible and powerful tool for modeling data distribution. It has the form of Gibbs distribution as $p_{\theta}(\mathbf{x}) = \exp(-E(\mathbf{x}))/A$, where $p_{\theta}(\mathbf{x})$ is the model distribution and A denotes the normalization constant. The computation of such probability is intractable due to the high cardinality of the data space. Recently, great progress has been made in solving this intractable function, including contrastive divergence [50], noise contrastive estimation [79], and score matching (SM) [104, 220, 221]. For example, SM solves this by first introducing the concept *score*, the gradient of the log-likelihood with respect to the data, and then matching the model score with the data score using Fisher divergence. This approach has been further improved by combining SM with denoising auto-encoding, forming the promising denoising score matching (DSM) strategy [248]. In this work, we will explore the potential of leveraging DSM for molecule geometry representation learning. We aim to utilize pairwise distance information, one of the most fundamental factors in the geometric molecule data.

Problem setup. Our goal here is to apply a self-supervised pretraining algorithm on a large molecular geometric dataset and adapt the pretrained representation for fine-tuning on geometric downstream tasks. For both the pretraining and downstream tasks, only the 3D geometric information is available, and our solution is agnostic in terms of the backbone geometric neural network.

4. Method

This section first introduces the GeoSSL framework and then proposes the GeoSSL-DDM algorithm. We start with exploring the coordinate perturbation for molecular data in Section 4.1. Then we introduce a coordinate-aware mutual information (MI) maximization formula and turn it into a coordinate denoising framework in Section 4.2. Nevertheless, the coordinate denoising is non-trivial since it requires geometric data reconstruction, and we adopt the score matching for estimation, as proposed in Section 4.3. The ultimate training objective is discussed in Section 4.4.

4.1. Coordinate Perturbation for Geometric Data

The mainstream self-supervised learning community designs the pretraining task by defining multiple views from the data, and these views share common information to some degree. Thus, by designing generative or contrastive tasks to maximize the mutual information (MI) between these views, the pretrained representation can encode certain key information. This will make the representation more robust and more generalizable to downstream tasks. In our work, we propose GeoSSL-DDM, an SE(3)-invariant self-supervised learning (SSL) method for molecule geometric representation learning.

The 3D geometric information or the atomic coordinates are critical to molecular properties. We carry out an additional ablation study to verify this in Appendix B.2. Then based on this acknowledgment, we introduce a geometry perturbation, which adds small noises to the atom coordinates. For notation, following Section 3, we define the original geometry graph and an augmented geometry graph as two views, denoted as $\mathbf{g}_1 = (X_1, R_1)$ and $\mathbf{g}_2 = (X_2, R_2)$, respectively. The augmented geometry graph can be seen as a coordinate perturbation to the original graph with the same atom types, *i.e.*, $X_2 = X_1$ and $R_2 = R_1 + \epsilon$, where ϵ is drawn from a normal distribution.

4.2. Coordinate Denoising with MI Maximization Framework: GeoSSL

The two views defined above share certain common information. By maximizing the mutual information (MI) between them, we expect that the learned representation can better capture the geometric information and is robust to noises and thus can generalize well to downstream tasks. To maximize the MI, we turn to maximize the following lower bound on the two geometry views, leading to the geometric self-supervised learning framework, GeoSSL:

$$\mathcal{L}_{\text{GeoSSL}} \triangleq \frac{1}{2} \mathbb{E}_{p(\mathbf{g}_1, \mathbf{g}_2)} \left[\log p(\mathbf{g}_1 | \mathbf{g}_2) + \log p(\mathbf{g}_2 | \mathbf{g}_1) \right]. \quad (4.1)$$

In Equation (4.1), we transform the MI maximization problem into maximizing the summation of two conditional log-likelihoods. In addition, these two conditional log-likelihoods are in the mirroring direction, and such symmetry can reveal certain nice properties, *e.g.*, it highlights the equal importance and uncertainty of the two views and can lead to a more robust representation of the geometry.

To solve Equation (4.1), we adopt the energy-based model (EBM) for estimation. EBM has been acknowledged as a flexible framework for its powerful usage in modeling distribution over highly-structured data, like molecules [84, 136]. To adapt it for GeoSSL, the objective can be turned into:

$$\begin{aligned} \mathcal{L}_{\text{GeoSSL-EBM}} &= \frac{1}{2} \mathbb{E}_{p(\mathbf{g}_1, \mathbf{g}_2)} \left[\log p(R_1 | \mathbf{g}_2) \right] + \frac{1}{2} \mathbb{E}_{p(\mathbf{g}_1, \mathbf{g}_2)} \left[\log p(R_2 | \mathbf{g}_1) \right] \\ &= \frac{1}{2} \mathbb{E}_{p(\mathbf{g}_1, \mathbf{g}_2)} \left[\log \frac{\exp(f(R_1, \mathbf{g}_2))}{A_{R_1 | \mathbf{g}_2}} \right] + \frac{1}{2} \mathbb{E}_{p(\mathbf{g}_2, \mathbf{g}_1)} \left[\log \frac{\exp(f(R_2, \mathbf{g}_1))}{A_{R_2 | \mathbf{g}_1}} \right], \end{aligned} \quad (4.2)$$

where the $f(\cdot)$ are the negative of energy functions, and $A_{R_1 | \mathbf{g}_2}$ and $A_{R_2 | \mathbf{g}_1}$ are the intractable partition functions. The first equation in Equation (4.2) is because the two views share the same atom types. This equation can be treated as denoising the atom coordinates of one view from the geometry of the other view. In the following, we will explore how to use the score matching for solving the above EBM estimation problem, and further transform the coordinate-aware GeoSSL to the denoising distance matching as the final objective.

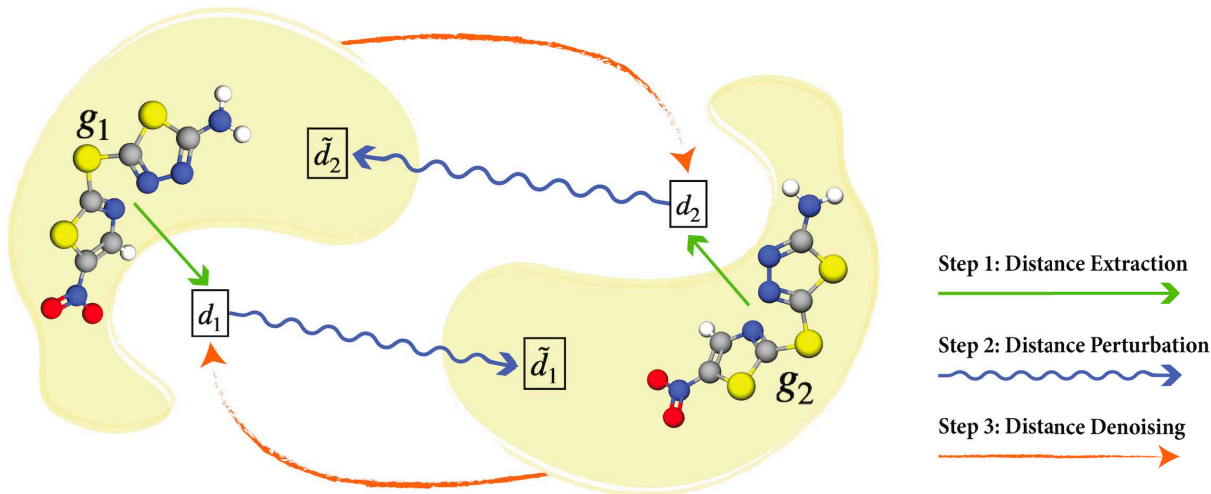


Figure 5. Pipeline for GeoSSL-DDM. The g_1 and g_2 are around the same local minima, yet with coordinate noises perturbation. Originally we want to conduct coordinate denoising between these two views. Then as proposed in GeoSSL-DDM, we transform it to an equivalent problem, *i.e.*, distance denoising. This figure shows the three key steps: extract the distances from the two geometric views, perform distance perturbation, and denoise the perturbed distances. Notice that the covalent bonds in the 3D data are added for illustration only.

4.3. From Coordinate Denoising to Distance Denoising: GeoSSL-DDM

Before going into details, first, we would like to briefly discuss denoising score matching (DSM). DSM has three main advantages that inspire us to apply it for solving the coordinate-aware GeoSSL. (1) The DSM solution has a nice formulation, such that the final objective function can be simplified with an intuitive explanation: GeoSSL-DDM can be seen as solving the denoising pairwise distance at multiple noise levels. (2) The score defined in geometric data can be viewed as a coordinate-based pseudo-force. Such pseudo-force can play an important role in the corresponding geometric representation learning. (3) In terms of the MI maximization, existing methods like InfoNCE [242], EBMs-NCE, and Representation Reconstruction [154] map the data to the *representation space* for either inter-data contrastive learning or intra-data reconstruction. This operation can avoid the decoding design issue for highly-structured data [51], yet the trade-off is losing the data-inherent information by a certain degree. In other words, the data-level reconstruction task (*e.g.*, DSM) is expected to lead to a more robust representation. Thus, considering the above points, we adopt DSM to our framework and propose GeoSSL-DDM. We expect that it can learn an expressive geometric representation function by solving the coordinate-aware GeoSSL. Additionally, the two terms in Equation (4.2) are in the mirroring direction. Thus in what follows, we adopt a proxy task that can calculate the two directions separately, and we take one for illustration, *e.g.*, $\log \frac{\exp(f(R_1, g_2))}{A_{R_1|g_2}}$.

4.3.1. Denoising Distance Matching. **Score.** The score is defined as the gradient of the log-likelihood w.r.t. the data, *i.e.*, the atom coordinates in our case. Because the normalization function is a constant regarding the data, it will disappear during the score calculation. To adapt it into our setting, the score is obtained as the gradient of the negative energy function w.r.t. the atom coordinates, as:

$$s(R_1, \mathbf{g}_2) \triangleq \nabla_{R_1} \log p(R_1 | \mathbf{g}_2) = \nabla_{R_1} f(R_1, \mathbf{g}_2). \quad (4.3)$$

If we assume that the learned optimal energy function, *i.e.*, $f(\cdot)$, possesses certain physical or chemical information, then the score in Equation (4.3) can be viewed as a special form of the pseudo-force. This may require more domain-specific knowledge, which we leave for future exploration.

Score decomposition: from coordinates to distances. Through back-propagation [209], the score on atom coordinates can be further decomposed into the scores attached to pairwise distances:

$$s(R_1, \mathbf{g}_2)_i = \sum_{j \neq i} \frac{\partial f(R_1, \mathbf{g}_2)}{\partial d_{1,ij}} \cdot \frac{\partial d_{1,ij}}{\partial r_{1,i}} = \sum_{j \neq i} \frac{1}{d_{1,ij}} \cdot s(\mathbf{d}_1, \mathbf{g}_2)_{ij} \cdot (r_{1,i} - r_{1,j}), \quad (4.4)$$

where $r_{1,i}$ is the i -th coordinate in \mathbf{g}_1 , $d_{1,ij}$ denotes the pairwise distance between the i -th and j -th nodes in \mathbf{g}_1 , and $s(\mathbf{d}_1, \mathbf{g}_2)_{ij} \triangleq \frac{\partial f(R_1, \mathbf{g}_2)}{\partial d_{1,ij}}$. Such decomposition has a nice intuition from the pseudo-force perspective: the pseudo-force on each atom can be further decomposed as the summation of pseudo-forces attached to the pairwise distances between this atom and all its neighbors. Note that here the pairwise atoms are connected in the 3D Euclidean space, not by the covalent bonds.

Denoising distance matching (DDM). Then we adopt the denoising score matching (DSM) [248] to our task. To be more concrete, we take the Gaussian kernel as the perturbed noise distribution on each pairwise distance, *i.e.*, $q_\sigma(\tilde{\mathbf{d}}_1 | \mathbf{g}_2) = \mathbb{E}_{p_{\text{data}}(\mathbf{d}_1 | \mathbf{g}_2)} [q_\sigma(\tilde{\mathbf{d}}_1 | \mathbf{d}_1)]$, where σ is the deviation in Gaussian perturbation. One main advantage of using the Gaussian kernel is that the following gradient of conditional log-likelihood has a closed-form formulation: $\nabla_{\tilde{\mathbf{d}}_1} \log q_\sigma(\tilde{\mathbf{d}}_1 | \mathbf{d}_1, \mathbf{g}_2) = (\mathbf{d}_1 - \tilde{\mathbf{d}}_1) / \sigma^2$, and the objective function of DSM is to train a score network to match it. This trick was first introduced in [248], and has been widely utilized in deep generative modeling tasks [218, 219].

To adapt to our setting, this is essentially saying that we want to train a score network, *i.e.*, $s_\theta(\tilde{\mathbf{d}}_1 | \mathbf{g}_2)$, to match the distance perturbation, or we can say it aims at matching the pseudo-force with the pairwise distances from the pseudo-force aspect. By taking the Fisher divergence as the discrepancy metric and the trick mentioned above, the estimation objective can be simplified to

$$D_F(q_\sigma(\tilde{\mathbf{d}}_1 | \mathbf{g}_2) || p_\theta(\tilde{\mathbf{d}}_1 | \mathbf{g}_2)) = \frac{1}{2} \mathbb{E}_{p_{\text{data}}(\mathbf{d}_1 | \mathbf{g}_2)} \mathbb{E}_{q_\sigma(\tilde{\mathbf{d}}_1 | \mathbf{d}_1, \mathbf{g}_2)} \left[\|s_\theta(\tilde{\mathbf{d}}_1, \mathbf{g}_2) - \frac{\mathbf{d}_1 - \tilde{\mathbf{d}}_1}{\sigma^2}\|^2 \right] + C. \quad (4.5)$$

For more detailed derivations, please refer to Appendix B.3. In this section, we turn the coordinate-aware GeoSSL framework into a distance perturbation matching problem, which is equivalent to denoising distance matching, *i.e.*, GeoSSL-DDM. The corresponding pipeline is illustrated in Figure 5.

4.3.2. SE(3)-Invariant Score Network Modeling. The objective function in Equation (4.5) is essentially doing the distance denoising. Since the distance is a type-0 feature [234], we simply design an SE(3)-invariant score network as $s_\theta(\cdot)$. For modeling $h(\cdot)$, we take an SE(3)-equivariant 3D geometric graph neural network as the geometric representation backbone model. Following the notations in Section 3 and \mathbf{g}_2 modeling, we have

$$h(\mathbf{g}_2)_i = \text{3D-GNN}(T(\mathbf{g}_2))_i, \quad h(\mathbf{g}_2)_{ij} = h(\mathbf{g}_2)_i + h(\mathbf{g}_2)_j, \quad (4.6)$$

for the atom-level and atom pairwise-level representation. Then we define the score network as:

$$s_\theta(\tilde{\mathbf{d}}_1, \mathbf{g}_2)_{ij} = \text{MLP}(\text{MLP}(\tilde{\mathbf{d}}_{1,ij}) \oplus h(\mathbf{g}_2)_{ij}), \quad (4.7)$$

where \oplus is the concatenation and MLP is the multi-layer perception. GeoSSL-DDM is agnostic to the backbone geometric representation function, and its main module is the score network in Equation (4.7). Thus, GeoSSL-DDM is an SE(3)-invariant [67] pretraining algorithm. Meanwhile, the type-0 distance can be modeled in a more expressive SE(3)-equivariant manner, and we leave that for future work.

4.4. Ultimate Objective

With the above score network modeling, we can formulate the ultimate objective function. We adopt the following four training tricks from [154, 218, 219] to stabilize the score matching training process. (1) We carry out the distance denoising at L -level of noises. (2) We add a weighting coefficient $\lambda(\sigma) = \sigma^\beta$ for each noise level, where β acts as the annealing factor. (3) We scale the score network by a factor of $1/\sigma$. (4) We sample the same atoms from the two geometry views with a masking ratio r . Ultimately, the objective function for GeoSSL-DDM, is as follows:

$$\begin{aligned} \mathcal{L}_{\text{GeoSSL-DDM}} = & \frac{1}{2L} \sum_{l=1}^L \sigma_l^\beta \mathbb{E}_{p_{\text{data}}(\mathbf{d}_1|\mathbf{g}_2)} \mathbb{E}_{q(\tilde{\mathbf{d}}_1|\mathbf{d}_1, \mathbf{g}_2)} \left[\left\| \frac{s_\theta(\tilde{\mathbf{d}}_1, \mathbf{g}_2)}{\sigma_l} - \frac{\mathbf{d}_1 - \tilde{\mathbf{d}}_1}{\sigma_l^2} \right\|_2^2 \right] \\ & + \frac{1}{2L} \sum_{l=1}^L \sigma_l^\beta \mathbb{E}_{p_{\text{data}}(\mathbf{d}_2|\mathbf{g}_1)} \mathbb{E}_{q(\tilde{\mathbf{d}}_2|\mathbf{d}_2, \mathbf{g}_1)} \left[\left\| \frac{s_\theta(\tilde{\mathbf{d}}_2, \mathbf{g}_1)}{\sigma_l} - \frac{\mathbf{d}_2 - \tilde{\mathbf{d}}_2}{\sigma_l^2} \right\|_2^2 \right]. \end{aligned} \quad (4.8)$$

Algorithm 1 GeoSSL-DDM pretraining

```
1: Input: A 3D geometry dataset and  $L$  levels of Gaussian noise.
2: Output: A pre-trained 3D representation function  $h(\cdot)$ .
3: for each 3D geometry graph  $g_1$  do
4:   Obtain  $g_2$  by adding Gaussian noises to atom coordinates in  $g_1$ .
5:   for each noise level  $l \in \{1, \dots, L\}$  do
6:     Add noise to the pairwise distance with  $\tilde{d}_1 = d_1 + \sigma_l, \tilde{d}_2 = d_2 + \sigma_l$ .
7:     Get the score  $s_\theta(\tilde{d}_1, g_2), s_\theta(\tilde{d}_2, g_1)$  with Equation (4.7) accordingly.
8:   end for
9:   Update 3D GNN representation function  $h(\cdot)$  using Equation (4.8).
10: end for
```

The algorithm is in Algorithm 1.

Comparison with score matching in generative modeling. We note that score matching has been widely used for generative modeling tasks. One of the main drawbacks in the generative setting is the long mixing time for MCMC sampling. However, our work aims at representation learning, so such a sampling issue will not affect our task. We further note that there also exists a series of works exploring the score matching for conformation generation [209]. However, their scores or pseudo-forces are attached to the 2D topology (the covalent bonds), while our work is for the pure geometric data and is attached to the pairwise distances defined in the 3D Euclidean space.

5. Experiments

In this section, we compare our method with nine 3D geometric pretraining baselines, including one randomly initialized, one supervised, and seven self-supervised approaches. For the downstream tasks, we adopt 22 tasks covering quantum mechanics prediction, force prediction, and binding affinity prediction. We provide all the experiment details and ablation studies in Appendix B.4.

5.1. Backbone Models

Our proposed GeoSSL-DDM is model-agnostic, and here we evaluate our method using one of the state-of-the-art geometric graph neural networks, PaiNN [208]. We carry out the exact same experiments on another backbone model, SchNet [206], and present the results in Appendix B.4.

PaiNN [208] is a follow-up work of SchNet [206]. It addresses the limitation of rotational equivariance in SchNet by embracing rotational invariance, attaining a more expressive 3D geometric model.

Other backbone models. First, we want to highlight that what we propose is a general solution and is agnostic to the backbone 3D geometric models. And in addition to the PaiNN model, we want to acknowledge that, recently, there have been several works along this research line, including but not limited to [22, 61, 61, 125, 159, 202, 214]. Yet, they may require large computation resources and may be infeasible (*e.g.*, out of GPU memory)

in our setting. The decision is made by considering the model performance, computation efficiency, and memory cost. For more benchmark results and detailed comparisons of the 3D geometric models, please check Appendix B.1.

5.2. Baselines and Pretraining Dataset

Pretraining dataset. The PubChemQC database is a large-scale database with around 4M molecules with 3D geometries, and it calculates both the ground-state and excited-state 3D geometries using DFT (density functional theory). Due to the high computational cost, only several thousand molecules can be processed every day, and this dataset takes years of effort in total. Following this, Molecule3D [268] takes the ground-state geometries from PubChemQC and transforms the data formats into a deep learning-friendly way. It also parses essential quantum properties for each molecule, including energies of the highest occupied molecular orbital (HOMO) and the lowest occupied molecular orbital (LUMO), the energy gap between HOMO-LUMO, and the total energy. For our molecular geometry pretraining, we take a subset of 1M molecules with 3D geometries from Molecule3D.

Self-supervised learning pretraining baselines. We first consider the four coordinate-MI-unaware SSL methods: (1) *Type Prediction* is to predict the atom type of masked atoms; (2) *Distance Prediction* aims to predict the pairwise distances among atoms; (3) *Angle Prediction* is to predict the angle among triplet atoms, *i.e.*, the bond angle prediction; (4) *3D InfoGraph* adopts the contrastive learning paradigm by taking the node-graph pair from the same molecule geometry as positive and negative otherwise. Next, following the coordinate-aware GeoSSL framework introduced in Equation (4.1), we include two contrastive and one generative SSL baselines. (5) *GeoSSL-InfoNCE* [242] and (6) *GeoSSL-EBM-NCE* [154] are the two widely-used contrastive learning loss functions, where the goal is to align the positive views and contrast the negative views simultaneously. Finally, (7) *GeoSSL-RR* (RR for Representation Reconstruction) [154] is a generative SSL that is a proxy to maximize the MI. RR is a more general form of non-contrastive SSL methods like BOYL [75] and SimSiam [30], and the goal is to reconstruct each view from its counterpart in the representation space. Following this, our proposed GeoSSL-DDM, can be classified as generative SSL for distance denoising.

Supervised pretraining baseline. We also compare our method with a supervised pretraining baseline. As aforementioned, the large-scale pretraining dataset uses the DFT to calculate the energy and extracts the most stable conformers with the lowest energies, which reveal the most fundamental properties of molecules in the 3D Euclidean space. Thus, such energies can be naturally adopted as supervised signals, and we take this as a supervised pretraining baseline.

Table 6. Downstream results on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for test. The evaluation is mean absolute error, and the best results are in **bold**.

Pretraining	Alpha ↓	Gap ↓	HOMO ↓	LUMO ↓	Mu ↓	Cv ↓	G298 ↓	H298 ↓	R2 ↓	U298 ↓	U0 ↓	Zpve ↓
–	0.048	44.50	26.00	21.11	0.016	0.025	8.31	7.67	0.132	7.77	7.89	1.322
Supervised	0.049	45.33	26.61	21.77	0.016	0.026	8.97	8.59	0.170	8.35	8.19	1.346
Type Prediction	0.050	47.28	30.56	23.18	0.016	0.024	9.32	9.10	0.163	8.94	8.60	1.357
Distance Prediction	0.063	47.62	29.18	22.40	0.019	0.045	12.02	12.31	0.636	11.76	12.22	1.840
Angle Prediction	0.056	47.36	29.53	22.61	0.018	0.027	10.23	10.13	0.143	9.95	9.70	1.643
3D InfoGraph	0.053	44.79	27.09	21.66	0.016	0.027	9.22	8.78	0.143	8.94	9.11	1.465
GeoSSL-RR	0.048	44.85	25.42	20.82	0.015	0.025	8.56	8.20	0.133	7.89	7.62	1.329
GeoSSL-InfoNCE	0.052	45.65	26.70	21.87	0.016	0.027	9.17	9.62	0.130	8.77	8.63	1.519
GeoSSL-EBM-NCE	0.049	44.18	26.29	21.46	0.015	0.026	8.56	8.13	0.126	8.01	7.96	1.447
GeoSSL-DDM (ours)	0.046	40.22	23.48	19.42	0.015	0.024	7.65	7.09	0.122	6.99	6.92	1.307

Table 7. Downstream results on 8 force prediction tasks from MD17. We take 1K for training, 1K for validation, and the number of molecules for test are varied among different tasks, ranging from 48K to 991K. The evaluation is mean absolute error, and the best results are in **bold**.

Pretraining	Aspirin ↓	Benzene ↓	Ethanol ↓	Malonaldehyde ↓	Naphthalene ↓	Salicylic ↓	Toluene ↓	Uracil ↓
–	0.556	0.052	0.213	0.338	0.138	0.288	0.155	0.194
Supervised	0.478	0.145	0.318	0.434	0.460	0.527	0.251	0.404
Type Prediction	1.656	0.349	0.414	0.886	1.684	1.807	0.660	1.020
Distance Prediction	1.434	0.090	0.378	1.017	0.631	1.569	0.350	0.415
Angle Prediction	0.839	0.105	0.337	0.517	0.772	0.931	0.274	0.676
3D InfoGraph	0.844	0.114	0.344	0.741	1.062	0.945	0.373	0.812
GeoSSL-RR	0.502	0.052	0.219	0.334	0.130	0.312	0.152	0.192
GeoSSL-InfoNCE	0.881	0.066	0.275	0.550	0.356	0.607	0.186	0.559
GeoSSL-EBM-NCE	0.598	0.073	0.237	0.518	0.246	0.416	0.178	0.475
GeoSSL-DDM (ours)	0.453	0.051	0.166	0.288	0.129	0.266	0.122	0.183

5.3. Downstream Tasks on Quantum Mechanics and Force Prediction

QM9 [192] is a dataset of 134K molecules consisting of 9 heavy atoms. It includes 12 tasks that are related to the quantum properties. For example, U0 and U298 are the internal energies at 0K at 0K and 298.15K respectively, and U298 and G298 are the other two energies that can be transferred from H298 respectively. The other 8 tasks are quantum mechanics related to the DFT process. MD17 [32] is a dataset on molecular dynamics simulation. It includes eight tasks, corresponding to eight organic molecules, and each task includes the molecule positions along the potential energy surface (PES), as shown in Figure 4. The goal is to predict the energy-conserving interatomic forces for each atom in each molecule position. We follow the literature [126, 159, 207, 208] of using 1K for training and 1K for validation, while the test set (from 48K to 991K) is much larger.

The results on QM9 and MD17 are displayed in Tables 6 and 7 respectively. From Tables 6 and 7, we can observe that most the pretraining baselines tested perform on par with or even worse than the randomly-initialized baseline. The top performing baseline is the representation reconstruction method (RR), which optimizes the coordinate-aware MI; it outperforms the other baselines on 5 out of 12 tasks in QM9 and 6 out of 8 tasks in MD17.

This implies the potential of applying generative SSL for maximizing this coordinate-aware MI. Promisingly, our proposed GeoSSL-DDM, achieves consistently improved performance on all 12 tasks in QM9 and 8 tasks in MD17. All these observations empirically verify the effectiveness of the distance denoising in GeoSSL-DDM, which models the most determinant factor in molecule geometric data.

5.4. Downstream Tasks on Binding Affinity Prediction

Atom3D [236] is a recently published dataset. It gathers several core tasks for 3D molecules, including binding affinity. The binding affinity prediction is to measure the strength of binding interaction between a small molecule to the target protein. Here we will model both the small molecule and protein with their 3D atom coordinates provided. We follow Atom3D in data preprocessing and data splitting. For more detailed discussions and statistics, please check Appendix B.4.

During the binding process, there is a cavity in a protein that can potentially possess suitable properties for binding a small molecule (ligand), and it is termed a pocket [222]. Because of the large volume of the protein, we follow [236] by only taking the binding pocket, where there are no more than 600 atoms for each molecule and protein pair. To be more concrete, we consider two binding affinity tasks. (1) The first task is ligand binding affinity (LBA). It is gathered from [252] and the task is to predict the binding affinity strength between a small molecule and a protein pocket. (2) The second task is ligand efficacy prediction (LEP). We have a molecule bounded to pockets, and the goal is to detect if the same molecule has a higher binding affinity with one pocket compared to the other one.

Results in Table 8 illustrate that, for the LBA task, two pretraining baseline methods fail to generalize to LBA (the loss gets too large), and all the other pretraining baselines cannot beat the randomly initialized baseline. For the LEP task, the supervised and two contrastive learning pretraining baselines stand out for both ROC and PR metrics. Meaningfully, for both tasks, GeoSSL-DDM is able to achieve promising improvement, revealing that modeling the local region around conformer with distance denoising can also benefit binding affinity downstream tasks.

5.5. Discussion: Connection with Multi-task Pretraining

In the above experiments, we test multiple self-supervised and supervised pretraining tasks separately. Yet, all these pretraining methods are not contradicted but could be complementary instead. Existing work has successfully shown the effect of combining them in various ways. For example, [99] shows that jointly doing supervised and self-supervised pretraining can augment the pretrained representation. [154, 217] prove that contrastive and

Table 8. Downstream results on 2 binding affinity tasks. We select three evaluation metrics for LBA: the root mean squared error (RMSD), the Pearson correlation (R_p) and the Spearman correlation (R_S). LEP is a binary classification task, and we use the area under the curve for receiver operating characteristics (ROC) and precision-recall (PR) for evaluation. We run cross validation with 5 seeds, and the best results are in **bold**.

Pretraining	LBA			LEP	
	RMSD ↓	R_P ↑	R_C ↑	ROC ↑	PR ↑
–	1.463 ± 0.06	0.572 ± 0.02	0.568 ± 0.02	0.675 ± 0.04	0.549 ± 0.05
Supervised	1.551 ± 0.08	0.539 ± 0.03	0.533 ± 0.03	0.696 ± 0.03	0.554 ± 0.03
Charge Prediction	2.316 ± 0.80	0.387 ± 0.11	0.400 ± 0.11	0.630 ± 0.05	0.557 ± 0.07
Distance Prediction	1.542 ± 0.08	0.545 ± 0.03	0.540 ± 0.03	0.521 ± 0.07	0.479 ± 0.07
Angle Prediction	–	–	–	0.545 ± 0.07	0.504 ± 0.07
3D InfoGraph	–	–	–	0.540 ± 0.03	0.469 ± 0.03
GeoSSL-RR	1.515 ± 0.07	0.545 ± 0.03	0.539 ± 0.03	0.654 ± 0.05	0.518 ± 0.06
GeoSSL-InfoNCE	1.564 ± 0.05	0.508 ± 0.03	0.497 ± 0.05	0.693 ± 0.06	0.571 ± 0.08
GeoSSL-EBM-NCE	1.499 ± 0.06	0.547 ± 0.03	0.534 ± 0.03	0.691 ± 0.05	0.603 ± 0.07
GeoSSL-DDM (ours)	1.451 ± 0.03	0.577 ± 0.02	0.572 ± 0.01	0.776 ± 0.03	0.694 ± 0.06

generative SSL pretraining methods can be learned simultaneously as a multi-task pretraining. In addition, in terms of the molecule-specific pretraining, [154] empirically verifies that 2D topology and 3D geometry views can share certain information, and maximizing their mutual information together with 2D topology SSL for pretraining is beneficial.

With these insights, we would like to claim that all of these points are worth exploring in the future, especially in the line of pretraining for molecular geometry. Because pretraining datasets often come with multiple quantum properties and the 2D molecular topology can be obtained heuristically. Yet as the first step to explore self-supervised learning using only the 3D geometric data (*i.e.*, without covalent bonds), our study here would like to leave multi-task pretraining for future exploration.

6. Conclusions and Future Directions

We proposed a novel coordinate denoising method, coined GeoSSL-DDM, for molecular geometry pretraining. GeoSSL-DDM leverages an SE(3)-invariant score matching strategy, under the GeoSSL framework, to successfully decompose its coordinate denoising objective into the denoising of pairwise atomic distances in a molecule, which then can be effectively computed and directly target the determinant factors in molecular geometric data. We empirically verified the effectiveness and robustness of our method, showing its superior performance to nine state-of-the-art pretraining baselines on 22 benchmarking geometric molecular property prediction and binding affinity tasks.

Our work opens up venues for multiple promising directions. First, from the machine learning perspective, we propose a general pipeline on using EBM for MI maximization on geometric data pretraining. Yet, there are more explorations on the success of EBM, like GFlowNet [14], and it would be interesting to explore how to combine it with molecular

geometric data along this systematic path. In addition, GeoSSL does not utilize the 2D structure (*i.e.*, covalent bonds for molecules), and it would be desirable to consider how to utilize the distance denoising together with the 2D topology information.

In terms of applications, our proposed GeoSSL-DDM is a general framework, and it can be naturally applied to other geometric data, such as point clouds and protein pretraining. In addition, our current goal is to perform denoising in the local region, yet it would be interesting to explore larger regions. From this aspect, the denoising can be viewed as recovering the molecular dynamics trajectory, and we would explore how generalizable this pretrained representation is to downstream tasks.

Third Article.

A Group Symmetric Stochastic Differential Equation Model for Molecule Multi-modal Pretraining

by

Shengchao Liu^{1,2*}, Weitao Du^{3*}, Zhiming Ma³, Hongyu Guo⁴, and Jian Tang^{1,5,6}

- (¹) Université de Montréal, Montréal, QC, Canada
- (²) Mila-Québec Artificial Intelligence Institute, Montréal, QC, Canada
- (³) University of Chinese Academy of Sciences, Beijing, China
- (⁴) National Research Council Canada, Ottawa, ON, Canada
- (⁵) HEC Montréal, Montréal, QC, Canada
- (⁶) Canadian Institute for Advanced Research, Toronto, ON, Canada

This article was published in International Conference on Machine Learning (ICML) 2023 .

The main contributions of Shengchao Liu for this articles are presented as follows:

- Propose the idea;
- Conduct the experiments;
- Write the paper.

Weitao Du contributed to algorithm implementation and paper writing; Zhiming Ma, Hongyu Guo, and Jian Tang helped with discussion and paper writing.

ABSTRACT. Molecule pretraining has quickly become the go-to schema to boost the performance of AI-based drug discovery. Naturally, molecules can be represented as 2D topological graphs or 3D geometric point clouds. Although most existing pertaining methods focus on merely the single modality, recent research has shown that maximizing the mutual information (MI) between such two modalities enhances the molecule representation ability. Meanwhile, existing molecule multi-modal pretraining approaches approximate MI based on the representation space encoded from the topology and geometry, thus resulting in the loss of critical structural information of molecules. To address this issue, we propose MoleculeSDE. MoleculeSDE leverages group symmetric (*e.g.*, SE(3)-equivariant and reflection-antisymmetric) stochastic differential equation models to generate the 3D geometries from 2D topologies, and vice versa, *directly* in the input space. It not only obtains tighter MI bound but also enables prosperous downstream tasks than the previous work. By comparing with 17 pretraining baselines, we empirically verify that MoleculeSDE can learn an expressive representation with state-of-the-art performance on 26 out of 32 downstream tasks.

Keywords: Multi-modal pretraining; group symmetry; SDE; SE(3)-equivariant; reflection anti-symmetric; drug discovery.

1. Introduction

Artificial intelligence (AI) for drug discovery has recently attracted a surge of research interest in both the machine learning and cheminformatics communities, demonstrating encouraging outcomes in many challenging drug discovery tasks [36, 69, 72, 94, 113, 115, 140, 148, 194]. These successes are primarily attributed to the informative representations of molecules.

Molecules can be naturally represented as topological graphs, where atoms and covalent bonds are the nodes and edges. Additionally, molecular 3D structures (a.k.a. *conformations*) can be treated as 3D geometric graphs, where the atoms are the point clouds in the 3D Euclidean space. Based on such two modalities, tremendous representation methods have been proposed in a supervised setting [36, 69]. Further, by leveraging a large number of molecule datasets curated [9, 96, 105, 268], *molecule pretraining strategies* [99, 148, 230] have proven their effectiveness in learning robust and expressive molecule representations. To this end, most such pertaining works focus on exploring the 2D topology modality, and typical algorithms include reconstructing the masked substructures [99, 140] and aligning the positive subgraph pairs and contrasting the negative pairs [227, 256, 275] simultaneously. Recently, there have also been successful explorations [112, 145, 278] on the 3D conformation pretraining, where the key idea is to reconstruct the masked distances or coordinates through a group symmetric reconstruction operation.

Nevertheless, despite its shown potential for forming high-quality representations, multi-modal pretraining over the molecular 2D topologies and 3D conformations has been under-explored. GraphMVP [153] is the first to build a unified multi-modal self-supervised learning

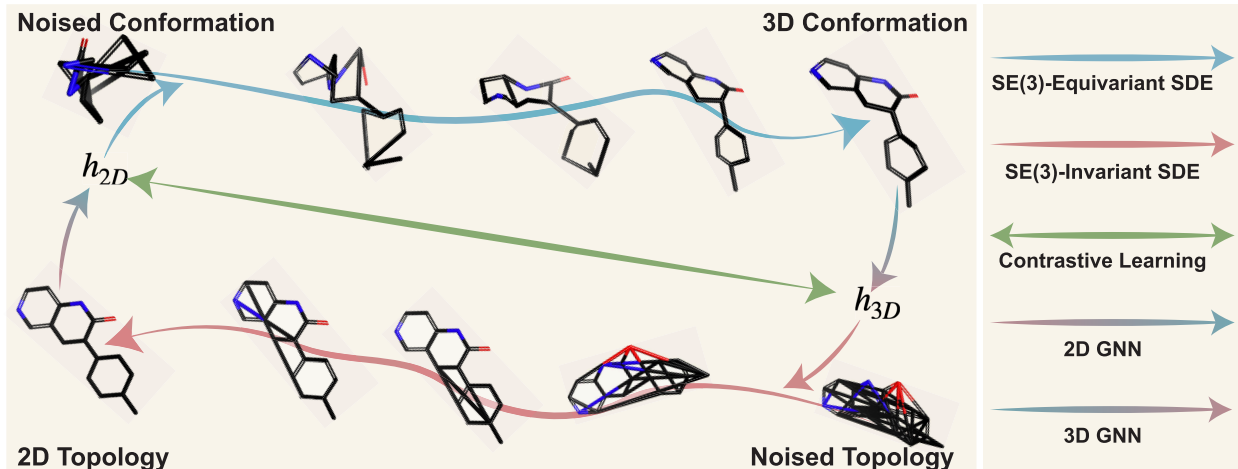


Figure 6. Illustration of the MoleculeSDE pretraining. It is composed of one contrastive learning and two generative learning objectives. Contrastive learning aims to align the 2D topological and 3D conformational representations for the same molecule. The two generative learning objectives are molecule conditional generation from 2D topology to 3D conformation and from 3D conformation to 2D topology, respectively. The generative modeling from topology to conformation is an SE(3)-equivariant diffusion process that satisfies the physical symmetry in molecule geometries. The other direction from conformation to topology is an SE(3)-invariant diffusion process since only the invariant type-0 features (nodes and edges) are considered. **We further include a demo in SI, showing the trajectory of this process.**

(SSL) paradigm. It introduces contrastive and generative learning to estimate the mutual information (MI) between the two modalities. Contrastive learning treats the topologies and conformations as positive if and only if they are referred to the same molecules. It aims to align the positive pairs while distributing the negative pairs. Such a contrastive idea has also been studied in [223]. More encouragingly, GraphMVP demonstrates the benefits of generative SSL, which aims at reconstructing the conformations from the topologies (or vice versa) by introducing a proxy to the evidence lower bound of MI estimation, *i.e.*, the *variational representation reconstruction (VRR)*. VRR transforms the reconstruction on the data space (*i.e.*, molecular topologies or conformations) to the representation space that compresses input features. Hence, the use of such a proxy can result in the loss of important structural information of molecules.

Our Approach. To address the aforementioned issue, we propose MoleculeSDE, a multi-modal pretraining method on molecules’ topologies and conformations. As illustrated in Figure 6, MoleculeSDE contains both contrastive and generative SSLs. The former adopts the EBM-NCE in [153], and the latter leverages the stochastic differential equation (SDE) framework [221]. Such design brings in two main benefits. First, for the objective function, the generative loss in GraphMVP is a proxy to the MI, while the diffusion process in MoleculeSDE leads to a more accurate MI estimation with less information loss. We list a brief performance comparison of two methods in Table 9 and will further provide theoretical

Table 9. Downstream tasks’ performance comparison with *merely* generative pretraining. The complete results are in Appendix C.8.

Model	Tox21 \uparrow	MUV \uparrow	Bace \uparrow	GAP \downarrow	U0 \downarrow	Aspirin \downarrow
VRR (GraphMVP)	73.6	75.5	72.7	44.64	13.96	1.177
SDE (MoleculeSDE)	75.6	80.1	79.0	42.75	11.85	1.087

insights that MoleculeSDE can lead to a more accurate MI estimation. Second, the SDE-based generative SSL enables prosperous downstream tasks. For example, MoleculeSDE enables conformation generation (CG) on tasks where only 2D topologies are available [263]. Based on this, we can apply more advanced geometric modeling methods for prediction. As shown in Section 5, such generated conformations lead to improved predictive performance over existing CG methods [49, 209].

The core components of the proposed MoleculeSDE are the two SDE generative processes. The first SDE aims to convert from topology to conformation. This conversion needs to satisfy the physical nature of molecules: the molecules’ physical and chemical attributes need to be equivariant to the rotations and translations in the 3D Euclidean space, *i.e.*, the **SE(3)-equivariant and reflection-antisymmetric** property(Appendix C.2), and we use **SE(3)-equivariance** for short. We note that existing topology to conformation deep generative methods are either SE(3)-invariant [209] or not SE(3)-equivariant [284]. We propose an SE(3)-equivariant diffusion process by building equivariant local frames. The inputs of local frames are SE(3)-equivariant vector features (*e.g.*, the atom coordinates), and the local frames transform them into three SE(3)-invariant features, which will be transformed back to the equivariant data using tensorization. The second SDE targets the conformation to topology reconstruction task on the discrete topological space. The main challenge for this task is to adopt the diffusion model for discrete data generation. We here follow the recent work on graph diffusion generation [114], where the joint generation of atom and bond leads to better estimation.

To the best of our knowledge, we are the first to build an **SE(3)-equivariant and reflection-antisymmetric** SDE for the topology to conformation generation and also the first to devise an SE(3)-invariant SDE for the conformation to topology generation for representation learning. We also note that our proposed MoleculeSDE is agnostic to the backbone representation methods since the SDE process is disentangled with the representation function, as illustrated in Figure 6.

Our main contributions include: (1) We propose a group symmetric pretraining method, MoleculeSDE, on the 2D and 3D modalities of molecules. (2) We provide theoretical insights on the tighter MI estimation of MoleculeSDE over previous works. (3) We show that MoleculeSDE enables prosperous downstream tasks. (4) We empirically verify that MoleculeSDE retains essential knowledge from both modalities, resulting in state-of-the-art performance on 26 out of 32 downstream tasks compared with 17 competitive baselines.

2. Related Work

Molecule SSL pretraining on a single modality. The pretraining on *2D molecular topology* shares common ideas with the general graph pretraining [230, 250]. One classical approach [99, 140] is to mask certain key substructures of molecular graphs and then perform the reconstruction in an auto-encoding manner. Another prevalent molecule pretraining method is contrastive learning [181], where the goal is to align the views from the positive pairs and contrast the views from the negative pairs simultaneously. For example, ContexPred [99] constructs views based on different radii of neighborhoods, Deep Graph InfoMax [247] and InfoGraph [227] treat the local and global graph representations as the two views, MolCLR [255] and GraphCL [274] create different views using discrete graph augmentation methods.

Recent studies start to explore the *3D geometric pretraining* on molecules. GeoSSL [145] proposes maximizing the mutual information between noised conformations using an SE(3)-invariant denoising score matching, and a parallel work [278] is a special case of GeoSSL using only one denoising layer. 3D-EMGP [112] is also a parallel work, but it is E(3)-equivariant, which needlessly satisfies the reflection-equivariant constraint in molecular conformation distribution.

Molecule SSL pretraining on multiple modalities. The GraphMVP proposes one contrastive objective (EBM-NCE) and one generative objective (variational representation reconstruction, VRR) to optimize the mutual information between the topological and conformational modalities. Specifically for VRR, it is a proxy loss to the evidence lower bound (ELBO) by doing the reconstruction in the representation space, which may risk losing information. 3D InfoMax [223] is a special case of GraphMVP, where only the contrastive loss is considered.

3. Preliminaries

2D topological molecular graph. A topological molecular graph is denoted as $g_{2D} = (\mathbf{X}, \mathbf{E})$, where \mathbf{X} is the atom attribute matrix and \mathbf{E} is the bond attribute matrix. The 2D graph representation with graph neural network (GNN) is:

$$\mathbf{H}_{2D} = \text{GNN-2D}(T_{2D}(g_{2D})) = \text{GNN-2D}(T_{2D}(\mathbf{X}, \mathbf{E})), \quad (3.1)$$

where T_{2D} is the data transformation on the 2D topology, and GNN-2D is the representation function. $\mathbf{H}_{2D} = [h_{2D}^0, h_{2D}^1, \dots]$, where h_{2D}^i is the i -th node representation.

3D conformational molecular graph. The molecular conformation is denoted as $g_{3D} = (\mathbf{X}, \mathbf{R})$, where $\mathbf{R} = \{\mathbf{r}^1, \mathbf{r}^2, \dots\}$ is the collection of 3D coordinates of atoms. The conformational representation is:

$$\mathbf{H}_{3D} = \text{GNN-3D}(T_{3D}(g_{3D})) = \text{GNN-3D}(T_{3D}(\mathbf{X}, \mathbf{R})), \quad (3.2)$$

where T_{3D} is the data transformation on the 3D geometry, and GNN-3D is the representation function. $\mathbf{H}_{3D} = [h_{3D}^0, h_{3D}^1, \dots]$, where h_{3D}^i is the i -th node representation. In our approach, we take the masking as the transformation for both 2D and 3D GNN, and the masking ratio is M . In what follows, we use \mathbf{x} and \mathbf{y} for the 2D and 3D graphs for notation simplicity, *i.e.*, $\mathbf{x} \triangleq g_{2D}$ and $\mathbf{y} \triangleq g_{3D}$.

SE(3)-Equivariance and Reflection-Antisymmetry. Two 3D geometric graphs \mathbf{R}_1 and \mathbf{R}_2 are SE(3)-isometric if there exists an element $g \in \text{SE}(3)$ such that $\mathbf{R}_2 = g\mathbf{R}_1$, where $g \in \text{SE}(3)$ is a 3D rotation or translation acting on each node (atom) of \mathbf{R}_1 . In this article, we will consider vector-valued functions defined on the 3D molecular graph. Specifically, given a conformation-related function $f(\mathbf{r}) : \mathbf{R}^3 \rightarrow \mathbf{R}^3$ on the graph \mathbf{R} , we say it’s equivariant if

$$f(g\mathbf{r}) = gf(\mathbf{r}) \tag{3.3}$$

for arbitrary $g \in \text{SE}(3)$. Since different chiralities can lead to different chemical properties [34, 37], the function $f(\mathbf{r})$ we consider in this article is SE(3)-equivariant and reflection-antisymmetric.

Stochastic Differential Equation (SDE). The score-based generative modeling with stochastic differential equations (SDEs) [221] provides a novel and expressive tool for distribution estimation. It is also a united framework including denoising score matching [218, 248] and denoising diffusion [92]. In general, these methods can be split into two processes: the forward and backward processes. The forward process is a parameter-free deterministic process, and it diffuses a data point \mathbf{x} into a random noise by adding noises, as

$$d\mathbf{x}_t = f(\mathbf{x}_t, t)dt + g(t)dw_t, \tag{3.4}$$

with $f(\mathbf{x}_t, t)$ the vector-value drift coefficient, $g(t)$ the diffusion coefficient, and w_t the Wiener process. Note that eq. (3.4) induces a family of densities $\mathbf{x}_t \sim p_t(\cdot)$. On the other hand, the backward process generates real data from the stationary distribution of Equation (3.4) by evolving along the following SDE:

$$d\mathbf{x}_t = [f(\mathbf{x}_t, t) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)]dt + g(t)dw_t. \tag{3.5}$$

Thus, the learning objective is to estimate $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$. This derivative term is called *score* in the literature of score matching [218, 221, 248]. We will use this framework in MoleculeSDE to generate 3D conformation and 2D topology.

4. The MoleculeSDE Method

In Section 4.1, we first provide the mutual information (MI) aspect in molecule multi-modal pretraining, and then we present the limitation of VRR. We discuss two SDE models as generative pretraining in Sections 4.2 and 4.3, respectively. The ultimate learning objective and inference process are illustrated in Section 4.4. Additionally in Section 4.5, we provide theoretical insights on how MoleculeSDE obtains a more accurate MI estimation.

4.1. An Overview from Mutual Information Perspective

Mutual information (MI) measures the non-linear dependency between random variables, and it has been widely adopted as the principle for self-supervised pretraining [91, 181]. The expectation is that, by maximizing the MI between modalities, the learned representation can keep the most shared information. Thus, MI-guided SSL serves as an intuitive and powerful framework for representation pretraining.

Recently, GraphMVP [153] transforms the MI maximization objective into the summation of two conditional log-likelihoods:

$$\mathcal{L}_{\text{MI}} = \frac{1}{2} \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [\log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x}|\mathbf{y})]. \quad (4.1)$$

GraphMVP [153] further solves Equation (4.1) by proposing a contrastive loss (EBM-NCE) and a generative loss (variational representation reconstruction, VRR). VRR conducts the reconstruction on the representation space, *i.e.*, from 2D (3D) data to the 3D (2D) representation space (details in Appendix C.6). The main advantage of VRR is its simple implementation without topology or conformation reconstruction, yet the trade-off is that it can lose critical information since it is only a proxy solution to generative learning.

Thus, to this end, we raise one question: *If there exists a more accurate conditional density estimation method for generative pretraining?* The answer is yes, and we propose MoleculeSDE. MoleculeSDE utilizes two stochastic differential (SDE) models to estimate Equation (4.1). SDE is a broad generative model class [116] where a neural network is used to model the score [220] of various levels of noise in a diffusion process [92]. To adapt it in MoleculeSDE, we propose an SDE from 2D topology to 3D conformation (Section 4.2) and an SDE from 3D conformation to 2D topology (Section 4.3). We also want to highlight that such reconstructions are challenging, as both the 2D topologies and 3D conformations are highly structured: the 2D topologies are permutation invariant, and the 3D conformations additionally obey the SE(3)-equivariance.

MoleculeSDE has three main advantages. (1) MoleculeSDE is a powerful generative pretraining method, as the SDE models have shown promising performance in applications including image generation [92, 221] and geometric representation [145, 278]. (2) MoleculeSDE is a more accurate estimation to Equation (4.1) than previous methods. Thus, the pretrained representation contains more critical information with a more accurate MI estimation. (3) MoleculeSDE enables prosperous downstream tasks, like topology to conformation generation for property prediction (Section 5.4).

4.2. An SE(3)-Equivariant Conformation Generation

The first objective we consider is the conditional generation from topology to conformation, $p(\mathbf{y}|\mathbf{x})$. One thing to highlight is that the molecule 3D conformation needs to satisfy the

physical property, *i.e.*, it needs to be equivariant to the rotation and translation in the 3D Euclidean space, which is known as the *SE(3)-equivariance and reflection-antisymmetry* property (Appendix C.2). Notice that for notation simplicity, we may call it *SE(3)-equivariance*, and only expand into details in the "Local frame" paragraph below.

The core module in SDE is the score network, $S_\theta^{2D \rightarrow 3D}$. To satisfy the physical nature of molecule 3D structure, such a score network needs to be SE(3)-equivariant. Specifically, the input includes the 2D graph \mathbf{x} , the noised 3D information \mathbf{y}_t at time t , and the time t . The output is the SE(3)-equivariant 3D scores at time t , accordingly. The goal is to use $S_\theta^{2D \rightarrow 3D}$ to estimate the score $\nabla \log p_t(\mathbf{y}_t|\mathbf{x})$.

To learn $p(\mathbf{y}|\mathbf{x})$, we formulate it as solving an SDE problem. Then based on the score network, the training objective is:

$$\mathcal{L}^{2D \rightarrow 3D} = \mathbb{E}_{\mathbf{x}, \mathbf{y}} \mathbb{E}_t \mathbb{E}_{\mathbf{y}_t | \mathbf{y}} \left[\left\| \nabla_{\mathbf{y}_t} \log p_t(\mathbf{y}_t | \mathbf{y}, \mathbf{x}) - S_\theta^{2D \rightarrow 3D}(\mathbf{x}, \mathbf{y}_t, t) \right\|_2^2 \right]. \quad (4.2)$$

Coordinate reconstruction. Notice that both 2D topologies and 3D conformations share the same atom information (atom types), so in this subsection specifically, by reconstructing \mathbf{y} , we are referring to reconstructing the coordinates \mathbf{R} . Thus, the objective function becomes:

$$\mathcal{L}_{2D \rightarrow 3D} = \mathbb{E}_{\mathbf{x}, \mathbf{R}} \mathbb{E}_t \mathbb{E}_{\mathbf{R}_t | \mathbf{R}} \left[\left\| \nabla_{\mathbf{R}_t} \log p_t(\mathbf{R}_t | \mathbf{R}, \mathbf{x}) - S_\theta^{2D \rightarrow 3D}(\mathbf{x}, \mathbf{R}_t, t) \right\|_2^2 \right], \quad (4.3)$$

Local frame. Before going into details of the score network, we want to introduce the SE(3)-equivariant & reflection-antisymmetric local frame (Appendix C.2). Such equivariant frames are introduced to fill in the gap between invariant 2D features and the output SE(3) vector field. It is equivalent to a 3D coordinate system that transforms equivariantly with the geometric graph. Through equivariant frames, we can project noised 3D coordinates into invariant scalars (isomers are projected differently), such that they are ready to be combined with invariant 2D features. On the other hand, by projecting back the 2D graph’s invariant predictions into an equivariant frame, our final output can transform equivariantly with respect to global rotation and translation. We leave the precise formulations of our local frames in Appendix C.2. Briefly in MoleculeSDE, we focus on the equivariant frame attached on each edge $(\mathbf{r}^i, \mathbf{r}^j)$:

$$\mathbf{t}_{\text{frame}}^{ij} = \text{Local-Frame}(\mathbf{r}^i, \mathbf{r}^j). \quad (4.4)$$

SE(3)-equivariant score network. Then we introduce how to build an SE(3)-equivariant score network based on the local frame. We first concat the atom representations h_{2D} into the atom pairwise representations, as $e_{2D}^{ij} = \text{MLP}(\text{concat}\{h_{2D}^i || h_{2D}^j\})$ for the i -th and j -th atoms. Then the 2D pairwise representations are further added to the 3D pairwise representations $e_{3D}^{ij} = \text{projection}_{\mathbf{t}_{\text{frame}}^{ij}}(\mathbf{r}^i, \mathbf{r}^j)$, produced by the equivariant frames. Then the final **invariant** edge feature e^{ij} is defined by:

$$e^{ij} = \text{rbf}(r^{ij}) \odot e_{2D}^{ij} + e_{3D}^{ij}, \quad (4.5)$$

where r^{ij} denotes the relative distance between the i -th and j -th atoms, and we use the radial basis function (RBF) to embed such distance features. Note that the input 3D coordinates and the corresponding distance matrix $\{r^{ij}\}$ are based on the diffused positions at a given diffusion step, rather than the ground truth 3D conformation.

Then we process e^{ij} through multiple graph attention layers [213]: $h^{ij} = \text{Attention}(e^{ij})$. Finally, by pairing the invariant aggregated edge features h^{ij} with our SE(3)-equivariant frames $\mathbf{t}_{\text{frame}}^{ij}$, we get the vector-valued score function: $S(\mathbf{r}^i) = \sum_j h^{ij} \odot \mathbf{t}_{\text{frame}}^{ij}$. Here, our equivariant construction guarantees that the output vector field is SE(3)-equivariant and reflection-antisymmetric.

Discussions. We want to clarify the following points between molecule geometric modeling (h_{3D}) and score network ($S_{\theta}^{2D \rightarrow 3D}$). (1) The score network proposed here is SE(3)-equivariant and reflection-antisymmetric. The input to the score network is the topology and diffused conformation, so it cannot be shared with the molecule geometric modeling, where the input is the ground-truth 3D conformation. (2) To the best of our knowledge, we are the first to propose the SE(3)-equivariant and reflection-antisymmetric SDE for the topology to conformation generation task.

4.3. An SE(3)-Invariant Topology Generation

The second objective is to reconstruct the 2D topology from 3D conformation, *i.e.*, $p(\mathbf{x}|\mathbf{y})$. Note that the 2D topology information (atoms and bonds) belongs to the type-0 feature [234], thus such a generative process should satisfy the SE(3)-invariance property. If we formulate it as an SDE problem, then the training objective is:

$$\mathcal{L}^{3D \rightarrow 2D} = \mathbb{E}_{\mathbf{y}, \mathbf{x}} \mathbb{E}_t \mathbb{E}_{\mathbf{x}_t | \mathbf{x}} \left[\left\| \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t | \mathbf{x}, \mathbf{y}) - S_{\theta}^{3D \rightarrow 2D}(\mathbf{y}, \mathbf{x}_t, t) \right\|_2^2 \right], \quad (4.6)$$

where $S_{\theta}^{3D \rightarrow 2D}$ is the score network.

SE(3)-invariant score network. For modeling $S_{\theta}^{3D \rightarrow 2D}$, it needs to satisfy the SE(3)-invariance symmetry property. The inputs are 3D conformational representation \mathbf{y} , the noised 2D information \mathbf{x}_t at time t , and time t . The output of $S_{\theta}^{3D \rightarrow 2D}$ is the invariant 2D score function at time t , as $\nabla \log p_t(\mathbf{x}_t | \mathbf{y})$. As introduced in Section 3, the diffused 2D information contains two parts: $\mathbf{x}_t = (\mathbf{X}_t, \mathbf{E}_t)$, so the corresponding forward SDE is a joint variant of Equation (3.4):

$$\begin{cases} d\mathbf{X}_t = f_{1,t}(\mathbf{X}_t, \mathbf{E}_t)dt + g_1(t)dw_t^1, \\ d\mathbf{E}_t = f_{2,t}(\mathbf{X}_t, \mathbf{E}_t)dt + g_2(t)dw_t^2, \end{cases} \quad (4.7)$$

where w_t^1 and w_t^2 are two independent Brownian motion. Then the score network $S_{\theta}^{3D \rightarrow 2D}$ is also decomposed into two parts for the atoms and bonds: $S_{\theta}^{\mathbf{X}_t}(\mathbf{x}_t)$ and $S_{\theta}^{\mathbf{E}_t}(\mathbf{x}_t)$.

Similar to the topology to conformation generation procedure, we first merge the 3D representation \mathbf{z}_y with the diffused atom feature \mathbf{X}_t as $H_0 = \text{MLP}(\mathbf{X}_t) + \mathbf{H}_y$. Then we apply a GCN as the score network to estimate the node-level score, as $S_\theta^{\mathbf{X}_t}(\mathbf{x}_t) = \text{MLP}(\text{concat}\{H_0 || \dots || H_L\})$, where $H_{i+1} = \text{GCN}(H_i, \mathbf{E}_t)$ and L is the number of GCN layer. On the other hand, the edge-level score is modeled by an unnormalized dot product attention $S_\theta^{\mathbf{E}_t}(\mathbf{x}_t) = \text{MLP}(\{\text{Attention}(H_i)\}_{0 \leq i \leq L})$.

4.4. Learning and Inference of MoleculeSDE

Learning. In addition to the two generative objectives, we also consider a contrastive loss, EBM-NCE [153]. EBM-NCE can be viewed as another way to approximate the mutual information $I(X; Y)$, and it is expected to be complementary to the generative SSL. Therefore, our final objective is

$$\mathcal{L}_{\text{MoleculeSDE}} = \alpha_1 \mathcal{L}_{\text{Contrastive}} + \alpha_2 \mathcal{L}^{2\text{D} \rightarrow 3\text{D}} + \alpha_3 \mathcal{L}^{3\text{D} \rightarrow 2\text{D}}, \quad (4.8)$$

where $\alpha_1, \alpha_2, \alpha_3$ are three coefficient hyperparameters.

Inference. After we train the SDE model from 2D topologies to 3D conformations, then we can generate 3D molecular structures out of a fixed 2D topology by ‘reversing’ the forward SDE. More precisely, we take the Predictor-Corrector sampling method [221] as tailored to our continuous framework. Further, generating the 3D conformations from 2D topologies enable us to conduct prosperous downstream tasks such as property prediction jointly with 2D and 3D data (**demo in SI**).

4.5. Theoretical Insights of MoleculeSDE

Since the two terms in Equation (4.1) are in the mirroring direction, here we take $\mathbf{x}|\mathbf{y}$ for theoretical illustrations. The other direction can be obtained similarly. We adapt the continuous diffusion framework proposed in [221], in which the Markov chain denoising generative model (DDPM) is included as a discretization of the continuous Ornstein–Uhlenbeck process. The diffusion framework originates from the noised score-matching scheme [248] of training the energy-based model (EBM), in which the authors introduced a noised version of $p(\mathbf{y})$ by adding noise to each data point $\tilde{\mathbf{y}} = \mathbf{y} + \epsilon$, where ϵ is sampled from a scaled normal distribution. Then,

$$\begin{aligned} & \min_{\theta} \mathbb{E}_{p(\tilde{\mathbf{y}})} \|\nabla_{\mathbf{y}} \log p(\tilde{\mathbf{y}}|\mathbf{x}) - \nabla_{\mathbf{y}} \log p_{\theta}(\tilde{\mathbf{y}}|\mathbf{x})\|_2^2 \\ & = \min_{\theta} \mathbb{E}_{p(\tilde{\mathbf{y}}|\mathbf{y})} \|\nabla_{\mathbf{y}} \log p(\tilde{\mathbf{y}}|\mathbf{y}, \mathbf{x}) - \nabla_{\mathbf{y}} \log p_{\theta}(\tilde{\mathbf{y}}|\mathbf{x})\|_2^2 + C, \end{aligned} \quad (4.9)$$

where the conditional score function $\nabla_{\mathbf{y}} \log p(\tilde{\mathbf{y}}|\mathbf{y}, \mathbf{x})$ is analytically tractable. The diffusion generative model further pushes the one-step (4.9) to a continuous noising process from raw data \mathbf{y} to \mathbf{y}_t for $0 \leq t \leq T$. We call \mathbf{y}_t the noising (forward) diffusion process starting at y ,

which is usually formulated as the solution of a stochastic differential equation. Then, the corresponding (continuous) score matching loss is:

$$\min_{\theta} \mathbb{E}_t \mathbb{E}_{\mathbf{y}} \mathbb{E}_{\mathbf{y}_t | \mathbf{y}} \|\nabla_{\mathbf{y}_t} \log p_t(\mathbf{y}_t | \mathbf{y}, \mathbf{x}) - \nabla_{\mathbf{y}_t} \log p_{\theta, t}(\mathbf{y}_t | \mathbf{x})\|_2^2. \quad (4.10)$$

It’s worth mentioning that the weighted continuous score matching is equivalent to learning the infinitesimal reverse of the noising process from t to $t + \Delta t$ for each time t , which greatly reduces the difficulty of recovering p_{data} from the white noise in one shot [92].

To make a connection between Equation (4.10) and Equation (4.1), it’s crucial to relate score matching with **maximal log-likelihood** method. To solve this problem, [102] defined a key quantity (ELBO) $\mathcal{E}_{\theta}^{\infty}(y|x)$ as a functional on the infinite-dimensional path space (consists of all stochastic paths starting at y). Then, the authors show that

$$\mathbb{E}_{p(y)} \log p_{\theta, T}(y|x) \geq \mathbb{E}_{p(y)} \mathcal{E}_{\theta}^{\infty}(y|x),$$

where the probability $p_{\theta, T}(y|x)$ corresponds to the marginal distribution of a parameterized (denoted by θ) SDE at time T . Moreover, the ELBO $\mathbb{E}_{p(y)} \mathcal{E}_{\theta}^{\infty}(y|x)$ is equivalent to the score matching loss. Therefore, training the diffusion model is equivalent to maximizing a lower bound of the likelihood defined by SDEs. Since the variational capacity of the infinite-dimensional SDE space is larger than previous models, we expect to find a better estimation of eq. (4.1).

5. Experiments

MoleculeSDE enables both a pretrained 2D and 3D representation and can be further fine-tuned toward the downstream tasks. Meanwhile, another main advantage of MoleculeSDE is that it also learns an SDE model from topology to conformation. Such a design enables us to adopt more versatile downstream tasks. For instance, there is a wide range of molecular property prediction tasks [263] considering only the 2D topology, yet the 3D conformation has proven to be beneficial towards such property prediction tasks [153]. Thus, with the pretrained generative model $p(\mathbf{y}|\mathbf{x})$, we can generate the corresponding 3D structure for each molecule topology and apply the pretrained 3D encoders, which is expected to improve the performance further. A visual illustration of such three categories of downstream tasks is in Figure 7.

5.1. Pretraining and Baselines

Dataset. For pretraining, we use PCQM4Mv2 [97]. It’s a sub-dataset of PubChemQC [174] with 3.4 million molecules with both the topological graph and geometric conformations. We are aware of the Molecule3D [269] dataset, which is also extracted from

Table 10. Results for molecular property prediction tasks (with 2D topology only). For each downstream task, we report the mean (and standard deviation) ROC-AUC of 3 seeds with scaffold splitting. The best and second best results are marked **bold** and **bold**, respectively.

Pre-training	BBBP \uparrow	Tox21 \uparrow	ToxCast \uparrow	Sider \uparrow	ClinTox \uparrow	MUV \uparrow	HIV \uparrow	Bace \uparrow	Avg \uparrow
– (random init)	68.1 \pm 0.59	75.3 \pm 0.22	62.1 \pm 0.19	57.0 \pm 1.33	83.7 \pm 2.93	74.6 \pm 2.35	75.2 \pm 0.70	76.7 \pm 2.51	71.60
AttrMask	65.0 \pm 2.36	74.8 \pm 0.25	62.9 \pm 0.11	61.2\pm0.12	87.7\pm1.19	73.4 \pm 2.02	76.8 \pm 0.53	79.7 \pm 0.33	72.68
ContextPred	65.7 \pm 0.62	74.2 \pm 0.06	62.5 \pm 0.31	62.2\pm0.59	77.2 \pm 0.88	75.3 \pm 1.57	77.1 \pm 0.86	76.0 \pm 2.08	71.28
InfoGraph	67.5 \pm 0.11	73.2 \pm 0.43	63.7 \pm 0.50	59.9 \pm 0.30	76.5 \pm 1.07	74.1 \pm 0.74	75.1 \pm 0.99	77.8 \pm 0.88	70.96
MolCLR	66.6 \pm 1.89	73.0 \pm 0.16	62.9 \pm 0.38	57.5 \pm 1.77	86.1 \pm 0.95	72.5 \pm 2.38	76.2 \pm 1.51	71.5 \pm 3.17	70.79
3D InfoMax	68.3 \pm 1.12	76.1 \pm 0.18	64.8 \pm 0.25	60.6 \pm 0.78	79.9 \pm 3.49	74.4 \pm 2.45	75.9 \pm 0.59	79.7 \pm 1.54	72.47
GraphMVP	69.4 \pm 0.21	76.2 \pm 0.38	64.5 \pm 0.20	60.5 \pm 0.25	86.5 \pm 1.70	76.2 \pm 2.28	76.2 \pm 0.81	79.8\pm0.74	73.66
MoleculeSDE (VE)	73.2\pm0.48	76.5\pm0.33	65.2\pm0.31	59.6 \pm 0.82	86.6 \pm 3.73	79.9\pm0.19	78.5\pm0.28	80.4\pm0.92	74.98
MoleculeSDE (VP)	71.8\pm0.76	76.8\pm0.34	65.0\pm0.26	60.8 \pm 0.39	87.0\pm0.53	80.9\pm0.37	78.8\pm0.92	79.5 \pm 2.17	75.07

Table 11. Results on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for testing. The evaluation is mean absolute error, and the best and the second best results are marked in **bold** and **bold**, respectively.

Pretraining	$\alpha \downarrow$	$\nabla \mathcal{E} \downarrow$	$\mathcal{E}_{\text{HOMO}} \downarrow$	$\mathcal{E}_{\text{LUMO}} \downarrow$	$\mu \downarrow$	$C_v \downarrow$	$G \downarrow$	$H \downarrow$	$R^2 \downarrow$	$U \downarrow$	$U_0 \downarrow$	ZPVE \downarrow
– (random init)	0.060	44.13	27.64	22.55	0.028	0.031	14.19	14.05	0.133	13.93	13.27	1.749
Type Prediction	0.073	45.38	28.76	24.83	0.036	0.032	16.66	16.28	0.275	15.56	14.66	2.094
Distance Prediction	0.065	45.87	27.61	23.34	0.031	0.033	14.83	15.81	0.248	15.07	15.01	1.837
Angle Prediction	0.066	48.45	29.02	24.40	0.034	0.031	14.13	13.77	0.214	13.50	13.47	1.861
3D InfoGraph	0.062	45.96	29.29	24.60	0.028	0.030	13.93	13.97	0.133	13.55	13.47	1.644
RR	0.060	43.71	27.71	22.84	0.028	0.031	14.54	13.70	0.122	13.81	13.75	1.694
InfoNCE	0.061	44.38	27.67	22.85	0.027	0.030	13.38	13.36	0.116	13.05	13.00	1.643
EBM-NCE	0.057	43.75	27.05	22.75	0.028	0.030	12.87	12.65	0.123	13.44	12.64	1.652
3D InfoMax	0.057	42.09	25.90	21.60	0.028	0.030	13.73	13.62	0.141	13.81	13.30	1.670
GraphMVP	0.056	41.99	25.75	21.58	0.027	0.029	13.43	13.31	0.136	13.03	13.07	1.609
GeoSSL-1L	0.058	42.64	26.32	21.87	0.028	0.030	12.61	12.81	0.173	12.45	12.12	1.696
GeoSSL	0.056	42.29	25.61	21.88	0.027	0.029	11.54	11.14	0.168	11.06	10.96	1.660
MoleculeSDE (VE)	0.056	41.84	25.79	21.63	0.027	0.029	11.47	10.71	0.233	11.04	10.95	1.474
MoleculeSDE (VP)	0.054	41.77	25.74	21.41	0.026	0.028	13.07	12.05	0.151	12.54	12.04	1.587

Table 12. Results on eight force prediction tasks from MD17. We take 1K for training, 1K for validation, and 48K to 991K molecules for the test concerning different tasks. The evaluation is mean absolute error, and the best results are marked in **bold** and **bold**, respectively.

Pretraining	Aspirin \downarrow	Benzene \downarrow	Ethanol \downarrow	Malonaldehyde \downarrow	Naphthalene \downarrow	Salicylic \downarrow	Toluene \downarrow	Uracil \downarrow
– (random init)	1.203	0.380	0.386	0.794	0.587	0.826	0.568	0.773
Type Prediction	1.383	0.402	0.450	0.879	0.622	1.028	0.662	0.840
Distance Prediction	1.427	0.396	0.434	0.818	0.793	0.952	0.509	1.567
Angle Prediction	1.542	0.447	0.669	1.022	0.680	1.032	0.623	0.768
3D InfoGraph	1.610	0.415	0.560	0.900	0.788	1.278	0.768	1.110
RR	1.215	0.393	0.514	1.092	0.596	0.847	0.570	0.711
InfoNCE	1.132	0.395	0.466	0.888	0.542	0.831	0.554	0.664
EBM-NCE	1.251	0.373	0.457	0.829	0.512	0.990	0.560	0.742
3D InfoMax	1.142	0.388	0.469	0.731	0.785	0.798	0.516	0.640
GraphMVP	1.126	0.377	0.430	0.726	0.498	0.740	0.508	0.620
GeoSSL-1L	1.364	0.391	0.432	0.830	0.599	0.817	0.628	0.607
GeoSSL	1.107	0.360	0.357	0.737	0.568	0.902	0.484	0.502
MoleculeSDE (VE)	1.112	0.304	0.282	0.520	0.455	0.725	0.515	0.447
MoleculeSDE (VP)	1.244	0.315	0.338	0.488	0.432	0.712	0.478	0.468

PubChemQC [174]. Yet, after confirming with the authors, certain mismatches exist between the 2D topologies and 3D conformations. Thus, in this work, we use PCQM4Mv2 for pretraining.

Baselines for 2D topology pretraining. Enormous 2D topological pretraining methods have been proposed [158, 160, 262, 265]. Recent works [230] re-explore the effects of

these pretraining methods, and we pick up the most promising ones as follows. AttrMask [99, 140], ContexPred [99], Deep Graph Infomax [247] and InfoGraph [227], MolCLR [255] and GraphCL [274]. The detailed explanations are in Section 2.

Baselines for 3D conformation pretraining. The 3D conformation SSL pretraining has been less explored. We adopt the comprehensive baselines from [145]. The type prediction, distance prediction, and angle prediction predict the masked atom type, pairwise distance, and triplet angle, respectively. The 3D InfoGraph predicts whether the node- and graph-level 3D representation are for the same molecule. RR, InfoNCE, and EBM-NCE are to maximize the MI between the conformation and augmented conformation using different objective functions, respectively. GeoSSL optimizes the same objective function using denoising score matching. Another work [278] is a special case of GeoSSL with one layer of denoising, and we name it GeoSSL-1L.

Baselines for 2D-3D multi-modality pretraining. There are two baselines on the 2D-3D multi-modal pretraining: vanilla GraphMVP [153] utilizes both the contrastive and generative SSL, and 3D InfoMax [223] only uses the contrastive learning part in GraphMVP.

Backbone models and MoleculeSDE. For all the baselines and MoleculeSDE, we use the same backbone models to better verify the effectiveness of the pretraining algorithms. We take the GIN model [266] and SchNet model [207] for modeling 2D topology and 3D conformation, respectively. For MoleculeSDE training, we consider both the Variance Exploding (VE) and Variance Preserving (VP) (details in Appendix C.5).

5.2. Downstream with 2D Topology

We consider eight binary classification tasks from MoleculeNet [263]. The results are in Table 10. We can observe that MoleculeSDE works best on 6 out of 8 tasks, and both the VE and VP version of MoleculeSDE pretraining can reach the best average performance.

5.3. Downstream with 3D Conformation

We consider 12 tasks from QM9 [192] and 8 tasks from MD17 [32]. QM9 is a dataset of 134K molecules consisting of 9 heavy atoms, and the 12 tasks are related to the quantum properties, such as energies at various settings. MD17 is a dataset on molecular dynamics simulation, and the 8 tasks correspond to 8 organic molecules. The goal is to predict the forces at different 3D positions. The results are in Tables 11 and 12, and MoleculeSDE can reach the best performance on 9 tasks in QM9 and 7 tasks in MD17.

5.4. Downstream with Topology to Conformation

We note that the pretraining in MoleculeSDE does two things: representation learning and topology/conformation generation. Such a conformation generation pretraining enables

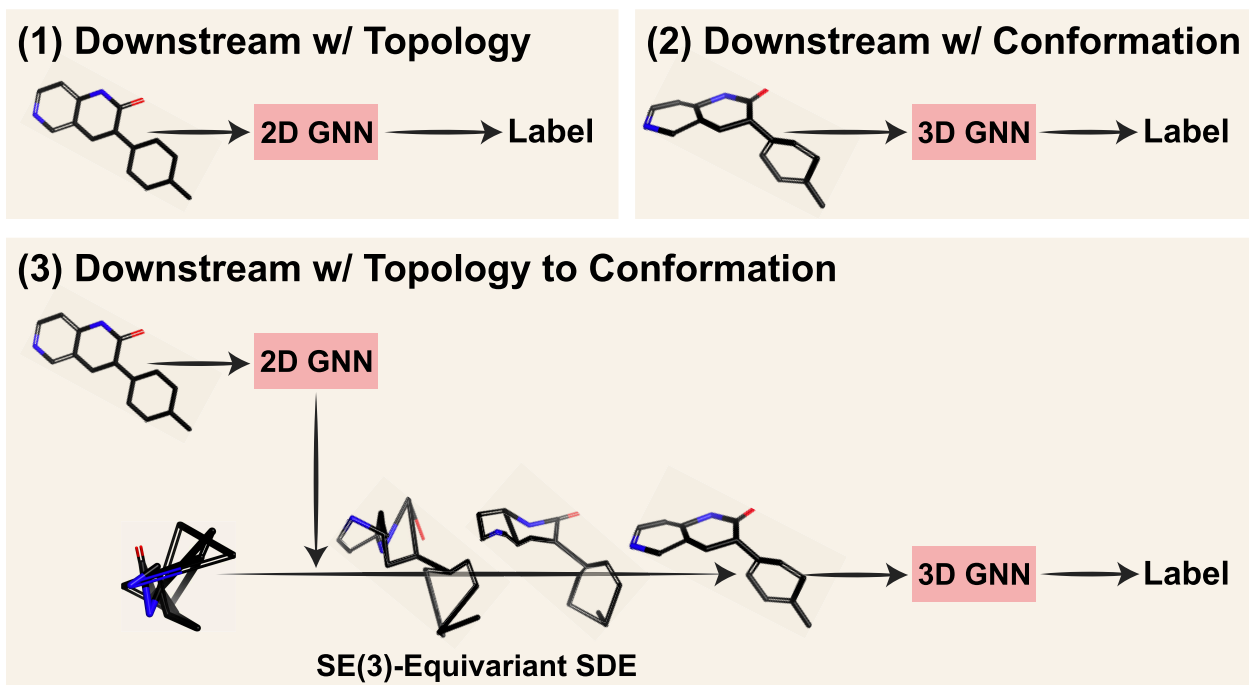


Figure 7. Illustration on three downstream tasks. The first two cover single-modal information only, and we fine-tune the pretrained 2D and 3D GNN from MoleculeSDE, respectively. The last downstream tasks contain topology only, then we use the pretrained 2D GNN and SE(3)-equivariant SDE model to generate conformations, followed by a 3D GNN for property prediction.

Table 13. Results for molecular property prediction with **SchNet** as backbone. The geometric structures (conformers) are generated using either MMFF or MoleculeSDE. We report the mean (and standard deviation) ROC-AUC of 3 seeds with scaffold splitting for each downstream task. CG denotes Conformation Generation.

Model	CG Method	BBBP \uparrow	Sider \uparrow	ClinTox \uparrow	Bace \uparrow
GIN	–	64.1 \pm 1.79	58.4 \pm 0.50	63.1 \pm 7.21	76.5 \pm 2.96
SchNet	MMFF	61.4 \pm 0.29	59.4 \pm 0.27	64.6 \pm 0.50	74.3 \pm 0.66
SchNet	ConfGF	62.7 \pm 1.97	60.1 \pm 0.87	64.1 \pm 2.83	73.2 \pm 3.53
SchNet	ClofNet	61.7 \pm 1.19	56.0 \pm 0.10	58.2 \pm 0.44	62.5 \pm 0.17
SchNet	MoleculeSDE	65.2\pm0.43	60.5\pm0.39	72.9\pm1.02	78.6\pm0.40

more prosperous downstream tasks. Here we consider 4 MoleculeNet tasks where only the 2D molecular graphs are available. Then we apply the SE(3)-equivariant conformation generation in MoleculeSDE, after which a 3D GNN will be trained for property prediction (Figure 7).

We consider three classic and state-of-the-art topology-to-conformation generation baselines. Merck molecular force field (MMFF) [81] is a heuristic method using physical simulation. ConfGF [209] is an SE(3)-invariant conformation generation method using score matching, and ClofNet [49] is the state-of-the-art SE(3)-equivariant conformation generation

method using GNN. With the generated conformations, we apply a SchNet model (without pretraining) for property prediction. We also add a 2D GNN baseline with the atom and bond type information. As shown in Table 13, we can observe that the CG baselines act worse than the 2D GNN for property prediction. Yet, MoleculeSDE can beat both the 2D and CG baselines. More discussions are in Appendix C.8.

5.5. Discussion on MoleculeSDE

For pretraining, data reconstruction is stronger than latent representation reconstruction. Starting from BOYL [75] and SimSiam [30], the non-contrastive SSL methods have been widely explored. GraphMVP [153] summarizes that these methods essentially reconstruct the latent representation space. Our proposed MoleculeSDE further proves that directly applying the data reconstruction is superior on the graph data. This observation also aligns well in the vision domain [86]. Complete results can be found in Appendix C.8.

6. Conclusion and Outlook

We proposed MoleculeSDE, a group symmetric pretraining method on the 2D topology and 3D geometry modalities of molecules. MoleculeSDE introduces the first SE(3)-equivariant and reflection-antisymmetric SDE for the topology to conformation generation and also the first SE(3)-invariant SDE for the conformation to topology generation for molecule representation learning. We provide theoretical insights that MoleculeSDE obtains tighter MI estimation over previous works. We also empirically verified that MoleculeSDE retains essential knowledge from both modalities, resulting in state-of-the-art performance on 26 out of 32 tasks compared to 17 competitive baselines.

We note that multi-modal pretraining has been widely explored in drug discovery, not only between the topologies and conformations (*i.e.*, the chemical structures), but also between the natural language and chemical structures. This research track is not exclusive to our work, and we believe that this can be a promising direction in the future exploration of the foundation model for molecule discovery.

Fourth Article.

Structured Multi-task Learning for Molecular Property Prediction

by

Shengchao Liu^{1,2*}, Meng Qu^{1,2}, Zuobai Zhang^{1,2}, Huiyu Cai^{1,2}, and Jian Tang^{1,3,4}

⁽¹⁾ Université de Montréal, Montréal, QC, Canada

⁽²⁾ Mila-Québec Artificial Intelligence Institute, Montréal, QC, Canada

⁽³⁾ HEC Montréal, Montréal, QC, Canada

⁽⁴⁾ Canadian Institute for Advanced Research, Toronto, ON, Canada

This article was published in Published in Artificial Intelligence and Statistics Conference (AISTATS) 2022.

The main contributions of Shengchao Liu for this articles are presented as follows:

- Propose the idea;
- Conduct the experiments;
- Write the paper.

Meng Qu, Zuobai Zhang, Huiyu Cai, and Jian Tang helped with discussion and paper writing.

ABSTRACT. Multi-task learning for molecular property prediction is becoming increasingly important in drug discovery. However, in contrast to other domains, the performance of multi-task learning in drug discovery is still not satisfying as the number of labeled data for each task is too limited, which calls for additional data to complement the data scarcity. In this paper, we study multi-task learning for molecular property prediction in a novel setting, where a relation graph between tasks is available. We first construct a dataset (ChEMBL-STRING) including around 400 tasks as well as a task relation graph. Then to better utilize such relation graph, we propose a method called SGNN-EBM to systematically investigate the structured task modeling from two perspectives. (1) In the *latent* space, we model the task representations by applying a state graph neural network (SGNN) on the relation graph. (2) In the *output* space, we employ structured prediction with the energy-based model (EBM), which can be efficiently trained through noise-contrastive estimation (NCE) approach. Empirical results justify the effectiveness of SGNN-EBM. Code is available on the GitHub repository.

Keywords: Multi-task learning; molecule representation; knowledge graph; EBM; drug discovery.

1. Introduction

Predicting the properties of molecules (*e.g.*, binding affinity with proteins, toxicity, ADME property) is a fundamental problem in drug discovery. Recently, we witness many successes of deep neural networks for molecular property prediction [2, 38, 99, 138, 147, 153, 195, 196, 200, 240, 263]. In particular, molecules are represented as molecular graphs, and graph neural networks [124]—which are neural network architectures specifically designed for graphs—are utilized for learning molecular representations. These neural networks are then usually trained with a set of labeled molecules. However, one big limitation for property prediction in drug discovery is that the labeled data are very limited, since they are very expensive and time-consuming to obtain. As a result, how to minimize the number of labeled data needed for effective molecular property prediction has long been a challenge in drug discovery.

One promising direction is multi-task learning, which tries to train multiple tasks (or properties) simultaneously so that the supervision or knowledge can be shared across tasks. Indeed, multi-task learning has been successfully applied to different domains and applications such as natural language understanding [215, 258], computer vision [164, 171], and speech recognition [107, 282]. In general, the essential idea of these works is to infer the relation among tasks. For example, [164] studied the hierarchical structure of different tasks; some more recent works [137, 258, 276] tried to infer the pairwise relation between tasks based on the gradients or loss of the tasks. There are also some recent work on multi-task learning for molecular property prediction [38, 137, 147, 195, 196, 263], which have shown very promising results. However, drug discovery possesses certain attributes distinguishable

from other domains, making it more challenging and interesting. (1) There is rich information in chemistry and biology domain, *e.g.*, the task relation if we are referring molecules as data and corresponding biological effects as the tasks. Then the question is how to better utilize such domain knowledge. (2) The number of molecules for each task is comparatively small, and merging data from different tasks may lead to a severe data sparsity issue (an example in Section 4), which adds more obstacles for learning.

In this paper, we study multi-task learning for molecular property prediction in a different setting, where a relation graph between tasks is explicitly given via domain knowledge. We first construct a large-scale dataset called ChEMBL-STRING by combining the chemical database of bioactive molecules (ChEMBL [168]) and the protein-protein interaction graph (STRING [231]). Specifically, we define a binary classification task based on an assay in ChEMBL, which measures the biological effects of molecules over a set of proteins. The relationship between different tasks are defined according to the relation of their associated sets of proteins, which can be inferred according to the protein-protein interaction graph in STRING. Finally, we are able to construct a large-scale dataset with 13,004 molecules and 382 tasks, together with the corresponding task relation graph.

With this constructed dataset, we propose a **novel** research problem: *How to do structured multi-task learning with an explicit task relation graph?* Our proposed solution is SGNN-EBM, which models the structured task information in both the *latent* and *output* space. More specifically, a state graph neural network (SGNN) can learn effective task representations by utilizing the relation graph, where the learnt representations effectively capture the similarities between tasks in the latent space. However, given a molecule, its labels are predicted independently for each task, which ignores the task dependency, *i.e.*, the dependency in the output space. Therefore, we further introduce formulating multi-task learning as structured prediction [11] problem, and apply an energy-based model (EBM) to model the joint distribution of the labels in the task space. Our proposed solution, coined SGNN-EBM, combines the advantages of both by adopting SGNN into the energy function in EBM, which provides higher capacity for structured task modeling. As training SGNN-EBM is generally computationally expensive, we deploy the noise contrastive estimation (NCE) [79] for effective training, which trains a discriminator to distinguish the observed examples and examples sampled from a noise distribution.

Our major contributions include (1) To our best knowledge, we are the **first** to propose doing multi-task learning with an explicit task relation graph; (2) We construct a domain-specific multi-task dataset with relation graph for drug discovery; (3) We propose SGNN-EBM for task structured modeling in both the latent and output space; (4) We achieve consistently better performance using SGNN-EBM.

2. Related Work

In the multi-task learning (MTL) literature, there are two fundamental problems: (1) how to learn the relation among tasks, and (2) how to model the task relation once available. Existing works on MTL *merely* focus on the first question, which can be roughly classified into two categories: architecture-specific MTL and architecture-agnostic MTL.

Architecture-specific MTL aims at designing special architecture to better transfer knowledge between tasks. Fully-adaptive network [164] dynamically groups similar tasks in a hierarchical structure. Cross-stitch network [171] applies multiple cross-stitch units and Bypass network [196] manipulates the architecture to model task relation. One drawback is that as the number of tasks grows, the requirement of computation memory increases linearly, which limits their application to large-scale setting (w.r.t. the task number).

Architecture-agnostic MTL provides a more general solution by learning to balance the tasks numerically. It has two components: a shared representation module and multiple task-specific prediction modules. Based on this framework, several methods have been proposed to learn a global linear task coefficient according to the optimization process, such as the the uncertainty [119], and task gradients and losses [31, 147, 157]. The learnt linear vector is then applied on the task-specific predictors. Instead of learning such linear vector, one alternative approach is to learn the pairwise task relation. RMTL [137] first handles this by applying a reinforcement learning framework to reduce the gradient conflicts between tasks. PCGrad and GradVac [258, 276] follow the same motivation and use gradient projection. However, there is one drawback on the high computational cost, since the pair-wise computation grows quadratically with the number of tasks; thus they are infeasible for large-scale setting (w.r.t. the task number).

Molecular property prediction has witnessed certain successful applications with MTL [38, 137, 138, 147, 169, 195, 240, 263] in terms of the robust performance gain. Furthermore, [133] finds that similarity within a target group significantly affects the performance of MTL on molecular binding prediction, revealing the importance of utilizing the task relation in drug discovery. However, all the aforementioned MTL methods do not possess the knowledge of the task relation and thus the main focus is to learn it in an architecture-specific or architecture-agnostic manner. While in this work, the task relation is given, and our focus moves to how to better model the structured task information in the MTL setting.

3. Problem Definition & Preliminaries

3.1. Problem Definition

Molecular Graph and Property Prediction. In molecular property prediction tasks, each data point \mathbf{x} is a molecule, which can be naturally viewed as a *topological graph*, where

atoms and bonds are nodes and edges accordingly. For each molecule \mathbf{x} , we want to predict T biological or physical *properties* [263], where each property corresponds to one *task*. For notation, we want to predict $\mathbf{y} = \{y_0, y_1, \dots, y_{T-1}\}$ for each molecule \mathbf{x} . Each task corresponds to C classes if it is a classification problem; and specifically in this work, we will be targeting at the binary tasks, *i.e.*, $C = 2$ and $y_i \in \{0,1\}, \forall i \in \{0, 1, \dots, T-1\}$.

Multi-Task Learning (MTL). Due to the inherent data scarcity issue in drug discovery [98, 166, 195, 263], training an independent model for each task often yields inferior performance. In practice [166], a more effective and widely-adopted approach is *multi-task learning (MTL)*, which tries to optimize multiple tasks simultaneously.

Task Relation Graph. A *task relation graph* is $\mathcal{G} = (V, E)$, where V is the node set of tasks and E are the corresponding edges between tasks. Here we add a linkage between two tasks if they are closely related. Thus, this relation graph can effectively complement the information sparsity of the labeled data for different tasks. More information on the task relation graph \mathcal{G} will be introduced in Section 4.

Structured Task Modeling. In this paper, we propose a *novel* research problem for MTL: how to do *structured task modeling* when the task relation graph is explicitly provided. Specifically, given a molecular graph \mathbf{x} , our goal is to jointly predict its labels for T tasks $\mathbf{y} = \{y_0, y_1, \dots, y_{T-1}\}$ with a task relation graph \mathcal{G} . In other words, we aim to model $p(\mathbf{y}|\mathbf{x}, \mathcal{G})$.

3.2. Preliminaries

Graph Neural Network (GNN) is a powerful tool in modeling structured data, like molecular graph and task relation graph. [69] first proposes a general GNN framework called *message passing neural network (MPNN)*. Following this, recent works have explored how to model the complex structured data like molecular graph [43, 52, 141, 200, 272] and knowledge graph [124, 266]. Typically for the node-level prediction, GNN models predict the node labels independently, and this limits the learning power of GNN to model the joint distribution of labels.

Energy-Based Model (EBM) uses a parametric energy function $E_\phi(\mathbf{x}, \mathbf{y})$ to fit the data distribution [132]. The energy function induces a density function with the Boltzmann distribution. Formally, the probability of $p_\phi(\mathbf{y}|\mathbf{x})$ can be written as:

$$p_\phi(\mathbf{y}|\mathbf{x}) = \frac{\exp(-E_\phi(\mathbf{x}, \mathbf{y}))}{Z_\phi(\mathbf{x})}, \tag{3.1}$$

where $E_\phi(\mathbf{x}, \mathbf{y})$ is the energy function, with which EBM is allowed to model the structured output space. $Z_\phi(\mathbf{x}) = \sum_{\mathbf{y}' \in \mathcal{Y}} \exp(-E_\phi(\mathbf{x}, \mathbf{y}'))$ is the partition function. Here $\mathcal{Y} = \{0,1\}^T$ is the label space, and the partition function is computationally intractable due to the high cardinality in $|\mathcal{Y}| = 2^T$. We will discuss how to cope with this issue for learning and inference in Section 5.

4. Dataset with Explicit Task Relation

In this section, we describe ChEMBL-STRING construction, a molecular property prediction dataset together with an explicit task relation graph. The *task* here refers to a binary classification problem on a ChEMBL assay [168], which measures certain biological effects of molecules, *e.g.*, toxicity, inhibition or activation of proteins or whole cellular processes, etc. We focus on tasks that target at proteins (*i.e.*, the binding affinity-related tasks), since the existing protein-protein interaction (PPI) data source can serve for the task relation extraction.

Our ChEMBL-STRING dataset is based on the *Large Scale Comparison (LSC)* dataset proposed by [166], which is filtered from the ChEMBL-20 database [168]. We account for a subset of 725 tasks which are protein-targeting. For each of these tasks, we collect the UniProt IDs [35] of the targeted proteins and combine all of them into a UniProt ID set. We then query the STRING database [231] to obtain PPI scores for all pairs of proteins in the set. With the collected PPI scores, we then heuristically define the edge weights w_{ij} , *i.e.*, task relation score, for task t_i and t_j in the task relation graph to be $\max\{\text{PPI}(s_i, s_j) : s_i \in S_i, s_j \in S_j\}$, where S_i denotes the protein set of task t_i . Therefore, the task relation graph proposed has a high quality to reveal the actual pharmaceutical effects for the molecular drugs.

As the experiment-based LSC dataset is very sparsely-labeled - only 0.78% of elements of the molecule-task matrix have a label of *active* or *inactive*, we densify the molecule-task label matrix by iteratively filtering out molecules and tasks whose number of labels is lower than a certain threshold. By setting the threshold value to 10, 50 and 100, we obtain 3 benchmark datasets with different level of data sparsity. The statistics of the benchmark datasets are listed in Table 14, and more detailed dataset generation procedure can be found in Appendix D.1.

Table 14. Statistics about ChEMBL-STRING datasets with explicit task relation, filtered by 3 thresholds. Threshold means the number of non-missing labels for each molecule/task.

Threshold	# Molecules	# Tasks	Sparsity
10	13,004	382	5.76%
50	932	152	66.70%
100	518	132	92.87%

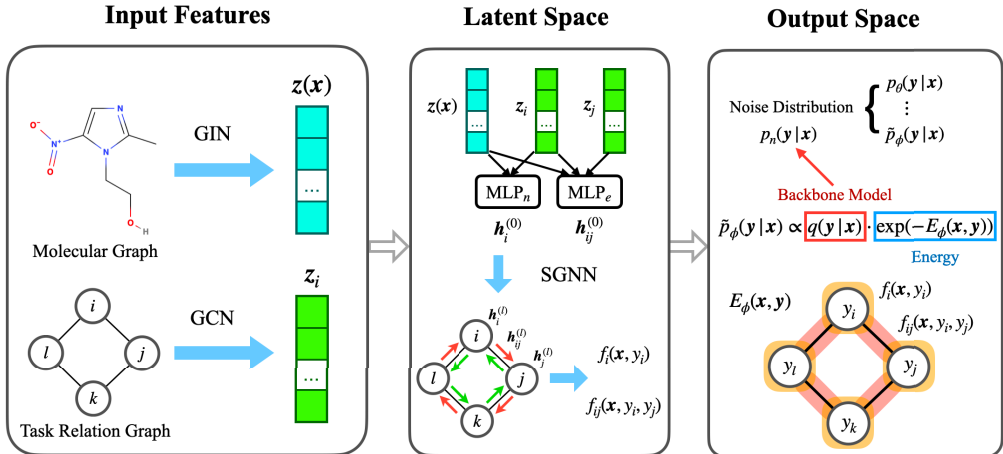


Figure 8. Pipeline of SGNN-EBM. We first obtain molecule and task embedding via GIN and GCN. Then, they are used to learn the latent representation for each task via a GNN model in the latent space. In SGNN-EBM, an SGNN model is used to model the task relation graph in the latent space and EBM learns the task distribution in the output space. The likelihood also applies the energy tilting term, which takes the same empirical distribution as the noise distribution for NCE.

5. Method: Structured Task Modeling

5.1. Overview

The mainstream multi-task learning (MTL) methods [31, 119, 147, 276] typically learn the task relation implicitly, which can guide to balance tasks during training. While in this paper, we focus on a novel setting where the *task relation graph* is explicitly given and the goal is to better model such relation graph. We first propose a dataset with an explicit task relation graph in Section 4, then in this section, we introduce two *structured MTL* approaches to modeling the task relation in the *latent* and *output* space respectively.

In the latent space, we propose to learn effective task representations with a State GNN (SGNN) on the task relation graph so that the learnt representations can capture the similarity between tasks. The property y_i in each task i can be independently predicted with the molecule information and its own task representation. More specifically, we can define the distribution as:

$$p_{\theta}(\mathbf{y}|\mathbf{x}, \mathcal{G}) = \prod_{i=0}^{T-1} p_{\theta}(y_i|\mathbf{x}, \mathcal{G}), \quad (5.1)$$

where $p_{\theta}(y_i|\mathbf{x}, \mathcal{G})$ is the prediction on the i -th task. We present this method in Section 5.2. More detailed description of GNN can be found in Appendix D.4.

One limitation of the SGNN is that it ignores the dependency between task labels y_i . To handle this issue, we further propose to model the task dependency in the output space and solve it under the energy-based model (EBM) framework, as a structured prediction

problem. The joint distribution of \mathbf{y} can be modeled with EBM as:

$$p_\phi(\mathbf{y}|\mathbf{x}, \mathcal{G}) = \frac{\exp(-E_\phi(\mathbf{x}, \mathbf{y}; \mathcal{G}))}{Z_\phi}, \quad (5.2)$$

where $E_\phi(\mathbf{x}, \mathbf{y}; \mathcal{G})$ is the energy function with flexible format. The noise contrastive estimation (NCE) is used to learn the EBM efficiently, and an outline of these methods is depicted in Figure 8.

Then we combine the advantages of both approaches by accounting the SGNN for energy function in EBM. Thus we are able to model the task relation in both the latent and output space, and we name this method as **SGNN-EBM** for solving structured MTL problems.

5.2. Modeling Task Relation in Latent Space

We propose *State GNN (SGNN)* to model the task relation in the latent space. The task relation is implicitly encoded in the learnt representations, and the final predictions are made independently for each task. We illustrate the pipeline of this model as follows.

Node- and Edge-Level Inputs. We first encode the molecules and tasks into the *embedding* space. For molecules, we adopt graph isomorphism network (GIN) [266], and the molecule embedding is $\mathbf{z}(\mathbf{x}) \in \mathbb{R}^{d_m}$, where d_m is the embedding dimension. Then for tasks, we use one-hot encodings (w.r.t. the task index) and pass them through a graph convolutional network (GCN) [124] to get task embedding: $\mathbf{z}^{(i)} \in \mathbb{R}^{d_t}, \forall i \in \{0, 1, \dots, T-1\}$, where d_t is the task embedding dimension. More details of GIN and GCN can be found in Appendices D.2 and D.3. Given the molecule and task embeddings, we will use them to construct the node- and edge-level inputs to SGNN as:

$$\begin{aligned} \mathbf{h}_i^{(0)}(\mathbf{x}) &= \text{MLP}_n^{(0)}(\mathbf{z}(\mathbf{x}) \oplus \mathbf{z}^{(i)}) \\ \mathbf{h}_{ij}^{(0)}(\mathbf{x}) &= \text{MLP}_e^{(0)}(\mathbf{z}(\mathbf{x}) \oplus \mathbf{z}^{(i)} \oplus \mathbf{z}^{(j)}), \end{aligned} \quad (5.3)$$

where \oplus is the concatenation of two tensors. $\text{MLP}_n^{(0)} : \mathbb{R}^{d_m+d_t} \rightarrow \mathbb{R}^{C \times d}$ and $\text{MLP}_e^{(0)} : \mathbb{R}^{d_m+2d_t} \rightarrow \mathbb{R}^{C \times C \times d}$ are two multi-layer perceptron (MLP) layers, operating on the node- and edge-level respectively. d is the dimension of the latent representation and $C = 2$ is the class number, and it also represents the states on each node and edge in SGNN. The node- and edge-level inputs in Equation (5.3) will then be fed to SGNN.

State GNN (SGNN). Different from the mainstream GNN models, SGNN has C and $C \times C$ states on each node and edge respectively, where each state delegates the representation for the corresponding label. Concretely, every node state represents the task w.r.t. the corresponding label, and edge state is composed of the pair-wise states from the two endpoint nodes. Thus, the representation for each node and edge state is defined as:

$$\begin{aligned} \mathbf{h}_i^{(0)}(\mathbf{x}, y_i) &= \mathbf{h}_i^{(0)}(\mathbf{x})[y_i] \\ \mathbf{h}_{ij}^{(0)}(\mathbf{x}, y_i, y_j) &= \mathbf{h}_{ij}^{(0)}(\mathbf{x})[y_i, y_j]. \end{aligned} \quad (5.4)$$

In this way, the representations of nodes and edges can well capture the information of each node label as well as the pairwise labels on an edge.

Such state-level view builds up the smallest granularity in SGNN. For example, during **message-passing propagation**, the key function in SGNN, only information with the same state will be exchanged between nodes and edges. Specifically, the propagation on the l -th layer is:

$$\begin{aligned} \mathbf{h}_i^{(l+1)}(\mathbf{x}, y_i) &= \text{MPNN}_n^{(l+1)}\left(\mathbf{h}_i^{(l)}(\mathbf{x}, y_i), \{\mathbf{h}_{ij}^{(l)}(\mathbf{x}, y_i, y_j) \mid \forall j, y_j\}\right) \\ \mathbf{h}_{ij}^{(l+1)}(\mathbf{x}, y_i, y_j) &= \text{MPNN}_e^{(l+1)}\left(\mathbf{h}_i^{(l)}(\mathbf{x}, y_i), \mathbf{h}_j^{(l)}(\mathbf{x}, y_j), \mathbf{h}_{ij}^{(l)}(\mathbf{x}, y_i, y_j)\right), \end{aligned} \tag{5.5}$$

where MPNN stands for the message-passing neural network layer [69]. MPNN_n is doing node aggregation by gathering information from edges with the same node state y_i ; MPNN_e stores the messages for each state pair (y_i, y_j) with the corresponding state information from the nodes. After repeating Equation (5.5) L times, we obtain the latent representation for each task given the molecule.

Independent Label Prediction. Finally, we make predictions for each task independently as Equation (5.1). For each task i , we first get the node representation by concatenating the two state representations, after which we apply a readout function R :

$$f_i(\mathbf{x}) = R(\{\mathbf{h}_i^{(l)}(\mathbf{x}, 0) \oplus \mathbf{h}_i^{(l)}(\mathbf{x}, 1) \mid l = 1, \dots, L\}), \tag{5.6}$$

where $R : \mathbb{R}^{2dL} \rightarrow \mathbb{R}$ is an MLP layer. Because $C = 2$ is the binary classification, the label distribution is defined via a sigmoid function, *i.e.*, $p(y_i = 1 | \mathbf{x}, \mathcal{G}) = \text{sigmoid}(f_i(\mathbf{x}))$. The loss function is the binary cross entropy function over all T tasks:

$$\mathcal{L} = \sum_{i=0}^{T-1} \log p(y_i | \mathbf{x}, \mathcal{G}). \tag{5.7}$$

Despite the effectiveness of learning task representations, SGNN fails to directly model the task dependency when making predictions as different task labels are predicted separately. To address this issue, we next propose a general method for modeling the task dependency under the structured prediction framework, which is able to predict task labels collectively to improve the result.

5.3. Modeling Task Relation in Output Space

The aforementioned MTL methods are predicting each task independently. However, there also exists a task distribution in the output space, *i.e.*, $p(\mathbf{y} = y_0, y_1, \dots, y_{T-1} | \mathbf{x})$. In this subsection, we propose to apply an energy-based model (EBM) to inject the prior knowledge about task dependency and model it with joint task distribution.

We define the **energy function** as the summation of first-order (node) and second-order (edge) factors on the graph:

$$E_\phi(\mathbf{x}, \mathbf{y}) = - \sum_{i=0}^{T-1} f_i(\mathbf{x}, y_i) - \lambda \sum_{\langle i, j \rangle \in \mathcal{G}} f_{ij}(\mathbf{x}, y_i, y_j), \quad (5.8)$$

where λ is a weighting coefficient. Thus the conditional probability under the EBM framework is defined as:

$$p_\phi(\mathbf{y}|\mathbf{x}) = \frac{\exp\left(\sum_i f_i(\mathbf{x}, y_i) + \sum_{ij} f_{ij}(\mathbf{x}, y_i, y_j)\right)}{Z_\phi}. \quad (5.9)$$

Activation Function. We apply the activation function $\sigma(\cdot) = \log(\text{softmax}(\cdot))$ on the first- and second-order factors. Then the readout function is $\tilde{R}(\cdot) = \log(\text{softmax}(\text{MLP}(\cdot)))$, where the softmax function is applied on the label/state space of each task and each task pair. The softmax function normalizes the scores of different label candidates, allowing us to compare them in the same range between 0 and 1. The logarithm function further scales the energy to 0 to ∞ , which is a common practice in EBM.

Energy Tilting Term. We have introduced EBM to model task relations in output space. However, directly training the energy-based model is still a challenging problem. To alleviate this issue, we leverage the energy tilting term from [4, 40, 179, 264], which takes EBM in the form of a correction or an exponential tilting of a pre-trained backbone model $q(\mathbf{y}|\mathbf{x})$. The pre-trained backbone model acts as a base model, and the energy function $\exp(-E_\phi(\mathbf{x}, \mathbf{y}))$ tries to tilt the base model for better results, yielding an integrated model as: $\tilde{p}_\phi(\mathbf{y}|\mathbf{x}) \propto q(\mathbf{y}|\mathbf{x}) \cdot \exp(-E_\phi(\mathbf{x}, \mathbf{y}))$, where the integrated model $\tilde{p}_\phi(\mathbf{y}|\mathbf{x})$ is named the *energy tilting distribution*. We will illustrate how to combine this energy tilting term in the learning and inference below.

5.4. SGNN-EBM

Then we will combine the structured modeling on both latent and output space together. As mentioned before, the energy function in EBM can have flexible formulation [132]; thus, we may as well parameterize it by adopting the node- and edge-level representation from SGNN. With minor modifications we have:

$$\begin{aligned} f_i(\mathbf{x}, y_i) &= \tilde{R}(\{\mathbf{h}_i^{(l)}(\mathbf{x}, y_i) \mid l = 1, \dots, L\}) \\ f_{ij}(\mathbf{x}, y_i, y_j) &= \tilde{R}(\{\mathbf{h}_{ij}^{(l)}(\mathbf{x}, y_i, y_j) \mid l = 1, \dots, L\}), \end{aligned} \quad (5.10)$$

where $\tilde{R} : \mathbb{R}^{dL} \rightarrow \mathbb{R}$ is a readout function defined as $\tilde{R} = \sigma(\text{MLP}(\cdot))$ and $\sigma(\cdot)$ is the activation function. Equation (5.10) is mapping the node and edge representations to scalars (or energies) indexed with the corresponding node and edge label.

As the number of message-passing layers L increases, the SGNN-based energy function (Equation (5.10)) can be seen as a general form to capture the higher-order dependency. However, according to the energy function decomposition in Equation (5.8), only first- and

second-order factors are considered during the EBM learning and inference. This discrepancy may raise some potential concern, and we carry on an ablation study in Section 6.3, where we empirically prove that slightly increasing L can be beneficial for the generalization performance. Yet, this is still worth further exploration in the future.

In the following sections, we will introduce how to do NCE learning and Gibbs sampling inference for our proposed SGNN-EBM model.

5.4.1. Learning. The learning process aims at optimizing ϕ to maximize the data likelihood. However, the problem is nontrivial as the partition function Z_ϕ is intractable. Our approach addresses this by using noise contrastive estimation (NCE) [79], which casts the problem of maximizing log-likelihood into a contrastive learning task. We first take the normalization constant Z_ϕ in Equation (3.1) as a learned scalar parameter. Then we transform the EBM learning into a binary classification problem by maximizing the following objective:

$$\mathcal{L}_{NCE} = \mathbb{E}_{\mathbf{y} \sim p_n} \log \frac{p_n(\mathbf{y}|\mathbf{x})}{p_n(\mathbf{y}|\mathbf{x}) + p_\phi(\mathbf{y}|\mathbf{x})} + \mathbb{E}_{\mathbf{y} \sim p_{\text{data}}} \log \frac{p_\phi(\mathbf{y}|\mathbf{x})}{p_n(\mathbf{y}|\mathbf{x}) + p_\phi(\mathbf{y}|\mathbf{x})}, \quad (5.11)$$

where p_{data} is the underlying data distribution, p_ϕ is the model distribution to approximate data distribution, and p_n is a noise distribution, whose samples serve as negative examples in the contrastive learning objective. Ideally, p_ϕ will be trained to approximate p_{data} for any noisy distribution. Yet in practice, the noise distribution should be close to the data distribution to facilitate the mining of hard negative samples. In addition [172], given an expressive energy function, we can fix $Z_\phi = 1$ and the resulting learned EBM will be self-normalized.

NCE with Tilting Term. The above objective function seems complicated. Nevertheless, it will become more concise as we combine the energy tilting term into NCE learning. We apply the backbone model for the noise distribution, *i.e.*, $p_n = q$, and replace the energy tilting term into Equation (5.11). With the self-normalized partition function, the NCE learning with energy tilting term can be written as:

$$\tilde{\mathcal{L}}_{NCE} = \mathbb{E}_{\mathbf{y} \sim p_n} \log \frac{1}{1 + \exp(-E_\phi(\mathbf{x}, \mathbf{y}))} + \mathbb{E}_{\mathbf{y} \sim p_{\text{data}}} \log \frac{1}{1 + \exp(E_\phi(\mathbf{x}, \mathbf{y}))}. \quad (5.12)$$

In this new objective function, we only need to draw samples from the noise distribution without computing their density, which is easy to operate. More detailed derivations are attached in Appendix D.6.

The Choice of Noise Distribution. One key component in NCE training is the choice of the noise distribution, p_n . NCE works for any given noise distribution, yet the algorithm empirically converges faster if the noise distribution p_n can stay close to the model distribution p_ϕ [220]. In the experiment, we propose two options for selecting the noise distributions. (1) We use a pre-trained model to be a *fixed* noise distribution, *e.g.*, the SGNN proposed in Section 5.2 and $p_n = p_\theta$. (2) We adopt an *adaptive* noise distribution, and start with a pre-trained model. The difference is that after training with this pre-trained noise

distribution for a few epochs, we will gradually update the noise distribution with our learned model, *i.e.*, updating p_n with the latest \tilde{p}_ϕ . The second idea aligns well with the curriculum learning [15], a learning process starting with easy data to hard data. Thus another way to interpret the adaptive noise distribution is that, we start with a simple distribution (from a pre-trained model distribution) and gradually using harder distribution (from the latest model distribution). We investigate the effect on the choices of noise distributions for NCE learning in the ablation study in Section 6.2.

Imputation for Missing Labels. For the SGNN-EBM training proposed in Section 5.4, we use the task distribution for predicting each data point, $p_\phi(\mathbf{y}|\mathbf{x})$, but some tasks do not have valid labels due to the label sparsity, as discussed in Sections 1 and 4. In SGNN-EBM, we propose to use the backbone model, q , to fill in the missing labels so as to calculate the probability. This strategy shares similar idea to the EM algorithm [175], which allows us to maximize a variational lower bound of the data likelihood. Empirically, experiment results help support this imputation strategy, yet, this is still work investigating in the future.

5.4.2. Inference. The inference procedure aims at computing the marginal distribution for each task, which can be further utilized for the label prediction for each task. The main challenge is how to calculate the intractable partition function during inference. We propose to approximate the distribution via Gibbs sampling [68]. Gibbs sampling is a classic MCMC-based inference method and the core idea is to generate samples by sweeping through each variable to a sample with the remaining variables fixed.

To adopt Gibbs sampling in our setting, for each data and T labels, $(\mathbf{x}, y_0, \dots, y_{T-1})$, we iteratively sample label for each task with other labels fixed. The update function at each iteration is:

$$p_\phi(y_i|\mathbf{y}_{-i}, \mathbf{x}) = \frac{\exp(f_i(\mathbf{x}, y_i) + \sum_{(i,j) \in \mathcal{G}} f_{ij}(\mathbf{x}, y_i, y_j))}{\sum_{y_i=0}^{C-1} \exp(f_i(\mathbf{x}, y_i) + \sum_{(i,j) \in \mathcal{G}} f_{ij}(\mathbf{x}, y_i, y_j))}, \quad (5.13)$$

where \mathbf{y}_{-i} denotes all T task labels except the i -th task. Then we take this as the tilting term, and apply $\tilde{p}(\mathbf{y}|\mathbf{x}) = p_\phi(\mathbf{y}|\mathbf{x}) \cdot q(\mathbf{y}|\mathbf{x})$ for sampling. To accelerate the convergence of Gibbs sampling, we take the backbone model for initial distribution.

6. Experiment Results

6.1. Main Results

Baselines. As described in Section 2, the memory cost of architecture-specific MTL methods (*e.g.*, Bypass network) is $O(T)$, and pair-wise architecture-agnostic MTL methods (RM TL [137], PCGrad [276], GradVac [258]) have $O(T^2)$ time complexity. Both are infeasible in the large-scale MTL setting (w.r.t. the number of tasks), so we exclude them in the experiments. For the baseline methods, we include standard single-task learning (STL),

Table 15. Main MTL results. All datasets are split into 8-1-1 for train, valid, and test respectively. For each method, we run 5 seeds and report the mean and standard deviation. The best performance is **highlighted**.

Method	p_n	ChEMBL 10	ChEMBL 50	ChEMBL 100
STL	–	71.67 ± 0.64	73.57 ± 1.20	70.81 ± 1.28
MTL	–	74.83 ± 0.61	79.37 ± 1.76	77.78 ± 1.59
UW	–	72.49 ± 0.53	79.68 ± 0.98	78.71 ± 1.93
GradNorm	–	75.17 ± 0.77	79.46 ± 1.27	78.75 ± 1.60
DWA	–	72.45 ± 1.31	79.35 ± 0.68	78.21 ± 2.31
LBTW	–	75.21 ± 0.49	79.52 ± 0.56	79.07 ± 0.99
SGNN	–	77.90 ± 0.88	79.67 ± 0.87	80.19 ± 0.67
SGNN-EBM	SGNN (Fixed)	78.04 ± 0.73	80.34 ± 1.08	80.48 ± 1.93
SGNN-EBM	SGNN (Adaptive)	78.35 ± 1.07	80.54 ± 1.02	81.15 ± 0.59

Table 16. The effect of different noise distributions p_n in NCE. Here all the noise distributions are fixed.

Method	p_n	ChEMBL-STRING 10	ChEMBL-STRING 50	ChEMBL-STRING 100
MTL	–	74.83 ± 0.61	79.37 ± 1.76	77.78 ± 1.59
UW	–	72.49 ± 0.53	79.68 ± 0.98	78.71 ± 1.93
GradNorm	–	75.17 ± 0.77	79.46 ± 1.27	78.75 ± 1.60
DWA	–	72.45 ± 1.31	79.35 ± 0.68	78.21 ± 2.31
LBTW	–	75.21 ± 0.49	79.52 ± 0.56	79.07 ± 0.99
SGNN	–	77.90 ± 0.88	79.67 ± 0.87	80.19 ± 0.67
SGNN-EBM	Uniform	58.66 ± 4.65	73.55 ± 0.61	75.49 ± 1.64
SGNN-EBM	MTL	75.71 ± 0.41	79.96 ± 1.41	78.41 ± 1.37
SGNN-EBM	UW	74.36 ± 0.87	80.26 ± 0.67	79.12 ± 1.79
SGNN-EBM	GradNorm	75.83 ± 0.73	80.18 ± 1.04	79.34 ± 1.31
SGNN-EBM	DWA	75.22 ± 1.16	80.18 ± 0.74	79.01 ± 1.94
SGNN-EBM	LBTW	76.16 ± 0.54	80.04 ± 0.50	79.68 ± 0.93
SGNN-EBM	SGNN	78.04 ± 0.73	80.34 ± 1.08	80.48 ± 1.93

standard multi-task learning (MTL), Uncertainty Weighing (UW) [119], GradNorm [31], Dynamic Weight Average (DWA) [157], and Loss-Balanced Task Weighting (LBTW) [147].

Our Methods. We first test SGNN, which *only* models the task relation graph in the latent space. On the other hand, EBM is very sensitive to the noise distribution, leading to unstable performance. Thus we will not test it separately as SGNN, and two following ablation studies can reveal more insights for it. Then we test our main proposal, SGNN-EBM. SGNN-EBM models the task relation graph in both the latent and output space under the EBM framework, where the energy function is defined as the SGNN. We explore two noise distributions in the NCE learning steps: (2.1) the first is a fixed pre-trained SGNN, $p_n = p_\theta$; (2.2) the second is taking the pre-trained SGNN, $p_n = p_\theta$, as initial noise distribution, and then adaptively updating this noise distribution with the latest model distribution $p_n = \tilde{p}_\phi$. More training details can be found in Appendix D.5.

Evaluation. We follow the mainstream evaluation metrics on MTL for drug discovery, *i.e.*, the mean of ROC-AUC over all T tasks. ROC-AUC is ranking-based, thus it can better match with the class-imbalance settings like molecular property prediction in drug discovery.

Table 17. The effect of layer number in SGNN, with 3 thresholds on ChEMBL-STRING.

# layer	10	50	100
0	77.45 \pm 1.03	80.63 \pm 0.80	80.82 \pm 2.09
2	77.56 \pm 1.00	80.78 \pm 0.85	81.13 \pm 2.04
4	76.98 \pm 0.91	80.42 \pm 0.82	81.06 \pm 2.09

Observation. We adopt the proposed dataset with three thresholds introduced in Section 4 for experiments. The main results are in Table 15. First we can see all the MTL methods are better than the STL, which matches with the common acknowledgement that the joint learning can improve the overall performance. Then for our proposed methods, we can see that modeling task relation in the latent space using SGNN reaches a good performance compared to all MTL baselines, while combining it with the EBM in the output space, *i.e.*, SGNN-EBM, can reach the best performance on all datasets. For the two SGNN-EBM models, they are consistently better than the SGNN model, while adaptively updated noise distribution can reach best performance. All these observations deliver an important message: structured task modeling is useful in MTL, and SGNN-EBM is an effective solution in achieving this goal.

6.2. Ablation Study 1: The Effect of p_n

In the NCE learning of EBMs, the performance highly depends on the noise distribution p_n . In Table 15 we show that the best method is SGNN-EBM with SGNN as both the energy function and noise distribution. Indeed we can take one uniform distribution and all pre-trained models (prior distribution) as the noise distribution, and we show that NCE-based structured prediction can obtain consistent performance gain when comparing to the corresponding prior distribution.

As in Table 16, the improvement by structured prediction is not huge but consistent on all datasets: for each pre-trained model, its SGNN-EBM counterpart can consistently improve the performance by taking it as a prior distribution in NCE learning. Such consistency consolidates the effectiveness of our solution.

6.3. Ablation Study 2: The Effect of L

We test SGNN-EBM* with $L = 0,2,4$ with all the other hyper-parameters fixed, where L is the number of layers in GNN. In the NCE learning, we are adapting the noise distributions from a pre-trained SGNN model, p_θ . The parameter L reflects that each node (molecule-task) in the graph aggregates features from its L -hop neighborhood.

As observed in Table 17, the SGNN-EBM improves the performance slightly with larger L in SGNN owing to the ability to model longer-term dependencies among labels. However,

as L increases, the performance will drop instead. One possible explanation is that the inference method, Gibbs Sampling, defined in Section 5.3 only considers first- and second-order factors, thus it fails to capture the long-term dependencies.

7. Conclusion and Future Direction

In this paper, we propose a novel research problem of MTL for molecular property prediction with an explicit task relation graph. We propose a novel approach to modeling the task relations in both the *latent* and *output* space. Experimental results demonstrate that SGNN-EBM outperforms competitive baselines.

We want to highlight that SGNN-EBM can fit to broad MTL problems, as long as the explicit task relation is accessible. But as the first step along this direction, we would like to start from a modest setting with assurance from the oracle, like explicit task relation from drug discovery domain. In addition, structured task modeling opens a new and promising research venue. For example, some MTL methods (RMTL [137], GradVac [258]) are able to extract the pairwise similarity to compose a task relation graph; yet, this view point is unexplored and would be interesting to combine with SGNN-EBM as the next step.

Fifth Article.

Multi-modal Molecule Structure-text Model for Text-based Editing and Retrieval

by

Shengchao Liu^{1,2}, Weili Nie³, Chengpeng Wang⁴, Jiarui Lu^{1,2}, Zhuoran Qiao⁵, Ling Liu⁶, Jian Tang^{1,7}, Chaowei Xiao^{3,8}, and Anima Anandkumar^{3,5}

- (¹) Université de Montréal, Montréal, QC, Canada
- (²) Mila-Québec Artificial Intelligence Institute, Montréal, QC, Canada
- (³) Nvidia Research, Santa Clara, CA, United States
- (⁴) University of Illinois Urbana-Champaign, Champaign, IL, United States
- (⁵) California Institute of Technology, Pasadena, CA, United States
- (⁶) Princeton University, Princeton, NJ, United States
- (⁷) HEC Montréal, Montréal, QC, Canada
- (⁸) Arizona State University, Tempe, AZ, United States

This article was published in Submitted to Nature Machine Intelligence (Under Review).

The main contributions of Shengchao Liu for this articles are presented as follows:

- Propose the idea;

- Conduct the experiments;
- Analyze the results;
- Visualize the pipelines;
- Write the paper.

Weili Nie, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Anima Anandkumar contributed to the idea discussion and paper writing; Chengpeng Wang helped conduct the ablation studies and visual analysis of molecule editing results; Jiarui Lu helped visualize the docking results.

ABSTRACT. There is increasing adoption of artificial intelligence in drug discovery. However, existing works use machine learning to mainly utilize the chemical structures of molecules yet ignore the vast textual knowledge available in chemistry. Incorporating textual knowledge enables us to realize new drug design objectives, adapt to text-based instructions, and predict complex biological activities. We present a multi-modal molecule structure-text model, MoleculeSTM, by jointly learning molecule’s chemical structures and textual descriptions via a contrastive learning strategy. To train MoleculeSTM, we construct the largest multi-modal dataset to date, namely PubChemSTM, with over 280K chemical structure-text pairs. To demonstrate the effectiveness and utility of MoleculeSTM, we design two challenging zero-shot tasks based on text instructions, including structure-text retrieval and molecule editing. MoleculeSTM possesses two main properties: open vocabulary and compositionality via natural language. In experiments, MoleculeSTM obtains the state-of-the-art generalization ability to novel biochemical concepts across various benchmarks.

Keywords: Foundation model for drug discovery; multi-modal modeling; molecule editing; molecule representation; drug discovery.

1. Introduction

Recent progress in artificial intelligence (AI) promises to be transformative for drug discovery [226]. AI methods have been used to augment and accelerate current computational pipelines [108, 115, 185], including but not limited to virtual screening [138, 198], metabolic property prediction [52, 142, 263], and targeted chemical structure generation and editing [106, 113, 152, 257].

Existing machine learning (ML) methods mainly focus on modeling the chemical structure of molecules through one-dimensional descriptions [127], two-dimensional molecular graphs [52, 142, 266], or three-dimensional geometric structures [6, 202, 207]. They also use supervised signals, *e.g.*, toxicity labels, quantum-mechanical properties, and binding affinity measurements. However, such a supervised setting requires expensive annotations on pre-determined label categories, impeding the application to unseen categories and tasks [110]. To overcome this issue, unsupervised pretraining on large-scale databases [105] has been proposed, with the main advantage being the ability to learn chemical structures

without supervised annotation by reconstructing the masked topological [99] or geometric [145] substructures. Compared to the supervised setting, although such pretrained models [99, 145] have proven to be more effective in generalizing to various downstream tasks by fine-tuning on a few labeled examples, it is still an open challenge to generalize unseen categories and tasks without such labeled examples or fine-tuning (*i.e.*, the so-called *zero-shot* setting [130] in ML). Additionally, existing molecule pretraining methods mostly incorporate only chemical structures, leaving the multi-modal representation less explored.

We have a vast amount of textual data that is human-understandable and easily accessible. This is now being harnessed in large-scale multi-modal models for images and videos [177, 184, 191, 193]. A natural language interface is an intuitive way to enable open vocabulary and description of tasks. Pretrained multi-modal models can generalize well to new categories and tasks, even in the zero-shot setting [177, 184, 191, 193]. They also enable agents to interactively learn to solve new tasks and explore new environments [57, 135]. We believe similar capabilities can also be obtained in molecular models by incorporating the vast textual knowledge available in the literature.

Previous work [280] has attempted to leverage the textual knowledge to learn the molecule representation. However, it only supports modeling with the 1D description (the simplified molecular-input line-entry system or SMILES) and learns the chemical structures and textual descriptions on a small-scale dataset (10K structure-text pairs). Furthermore, it unifies two modalities into a single language modeling framework and requires aligned data, *i.e.*, chemical structure and text for each sample, for training. As a result, it cannot adopt existing powerful pretrained models, and the availability of aligned data is extremely limited.

Our approach: We design a multi-modal foundation model for molecular understanding that incorporates both molecular structural information and textual knowledge. We demonstrate zero-shot generalization to new drug design objectives using text-based instructions and to the prediction of new complex biological activities without the need for labeled examples or fine-tuning.

We propose MoleculeSTM, consisting of two branches: the chemical structure branch and the textual description branch, to handle the molecules’ internal structures and external domain knowledge, respectively. Such a disentangled design enables MoleculeSTM to be integrated with the powerful existing models trained on each modality separately, *i.e.*, molecular structural models [106, 153] and scientific language models [13]. Given these pretrained models, MoleculeSTM bridges the two branches via a contrastive learning paradigm [153, 181].

To align such two branches with MoleculeSTM, we construct a structure-text dataset called PubChemSTM from PubChem [121], which is the largest multi-modal dataset to date in the community ($28\times$ larger than the existing dataset [280]). In PubChemSTM, each chemical structure is paired with a textual description, illustrating the chemical and physical properties or high-level bioactivities accordingly. Since MoleculeSTM is trained on

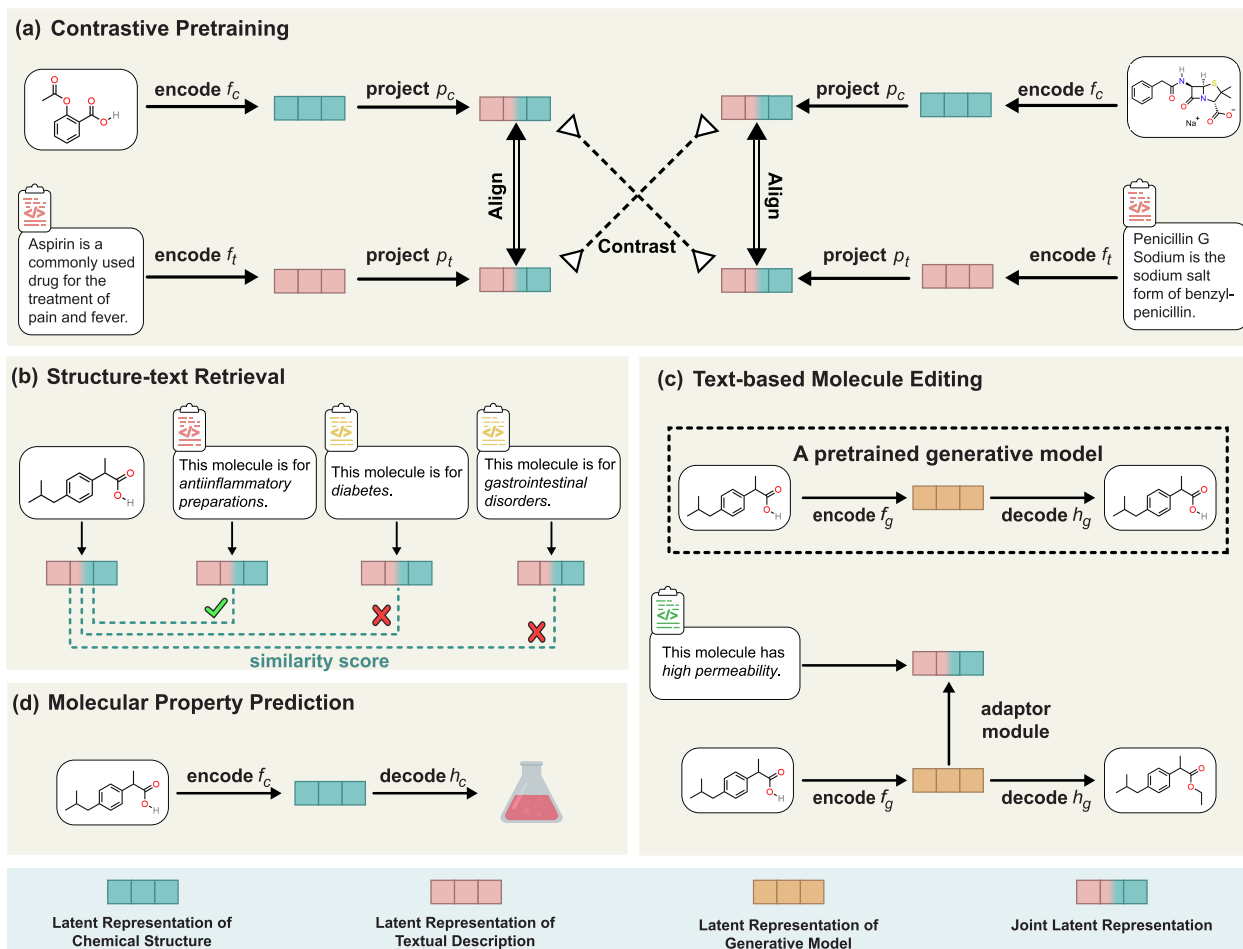


Figure 9. The pipelines of the pretraining task (a) and downstream tasks (b-d). (a) MoleculeSTM pretraining with two branches, the chemical structure (green) and textual description (pink). The contrastive learning strategy is adopted to align the structure-text pairs for the same molecules and contrast the structure-text pairs otherwise simultaneously in the joint latent space. (b) Zero-shot structure-text retrieval task to extract the textual description from the given chemical structure with the highest similarity score among $T = 3$ candidates. In the zero-shot setting, the pretrained encoders (f_c , f_t) and projectors (p_c , p_t) are used to calculate the similarity between the two branches on the joint latent space. (c) Zero-shot text-based molecule editing task to modify the input molecule based on the text prompt. A latent representation is optimized to balance its similarities with representations of the input molecule and a text prompt, which are obtained with a pretrained generative model and MoleculeSTM, respectively. The output molecule is then generated by decoding such optimized latent representation. (d) Molecular property prediction by end-to-end fine-tuning. Only the chemical structure is considered and the corresponding encoder is adopted from MoleculeSTM.

a large-scale structure-text pair dataset and such textual data contains open-ended chemical information, it can be generalized to diverse downstream tasks in a zero-shot manner.

To demonstrate the advantages of introducing the language modality, we design two challenging downstream tasks: the structure-text retrieval task and text-based molecule editing task, and we apply the pretrained MoleculeSTM on them in a zero-shot manner. By studying these tasks, we summarize two main attributes of MoleculeSTM: the open

vocabulary and compositionality. (1) Open vocabulary means our proposed MoleculeSTM is not limited to a fixed set of pre-defined molecule-related textual descriptions and can support exploring a wide range of biochemical concepts with the unbound vocabulary depicted by the natural language. In the drug discovery pipeline, such an attribute can be used for the text-based molecule editing in the lead optimization task and the novel disease-drug relation extraction in the drug re-purposing task. (2) Compositionality implies that we can express a complex concept by decomposing it into several simple concepts. This can be applied for the text-based multi-objective lead optimization task [103] where the goal is to generate molecules satisfying multiple properties simultaneously.

Empirically, MoleculeSTM reaches the best performance on six zero-shot retrieval tasks (up to 50% higher accuracy) and 20 zero-shot text-based editing tasks (up to 40% higher hit ratio) compared to the state-of-the-art methods. Furthermore, for molecular editing tasks, visual inspections reveal that MoleculeSTM can successfully detect critical structures implied in text descriptions. Additionally, we also explore whether MoleculeSTM can improve the performance on the standard molecular property prediction benchmark [263] via fine-tuning. Our results show that MoleculeSTM can achieve the best overall performance among nine baselines on eight property prediction tasks.

2. Results

2.1. Overview and Preliminaries

In this section, we first provide an overview of MoleculeSTM. Then, we introduce how to pretrain MoleculeSTM and apply the pretrained MoleculeSTM to three types of downstream tasks (Figure 9).

Overview. MoleculeSTM consists of two branches: the chemical structure branch and the textual description branch (\mathbf{x}_c and \mathbf{x}_t). The chemical structure branch illustrates the arrangement of atoms in a molecule. We consider two types of encoders f_c : Transformer [246] on the SMILES string and GNNs [52, 142, 266] on the 2D molecular graph. The textual description branch provides a high-level description of the molecule’s functionality, and we use the language model from a recent work [45] as the encoder f_t .

Pretraining. Within this design, MoleculeSTM aims to map the representations extracted from two branches to a joint space using two projectors (p_c and p_t) via contrastive learning [153, 181]. The essential idea of contrastive learning is to reduce the representation distance between the chemical structure and textual description pairs of the same molecule and increase the representation distance between the pairs from different molecules. Specifically, we initialize these two branch encoders with the pretrained single-modal

checkpoints [13, 106, 153] and then perform an end-to-end contrastive pretraining on collected dataset PubChemSTM, which consists of 281K chemical structure and text pairs.

Downstream: zero-shot structure-text retrieval. Given a chemical structure and T textual descriptions, the retrieval task is to select the textual description with the highest similarity to the chemical structure (or vice versa) based on a score calculated on the joint representation space. This is appealing for specific drug discovery tasks, such as drug re-purposing or indication expansion [1, 280]. We highlight that pretrained models are used for retrieval in the zero-shot setting, *i.e.*, without model optimization for this retrieval task.

Downstream: zero-shot text-based molecule editing. The objective of the molecule editing task is to modify the chemical structure of molecules such as functional group change [55] and scaffold hopping [19, 100]. Traditional methods for molecule editing highly rely on domain experts and could be subjective or biased [48, 71]. ML methods have provided an alternative strategy to solve this issue. Given a fixed pretrained molecule generative model (encoder f_g and decoder h_g), the ML editing methods learn a semantically meaningful direction on the latent representation (or latent code) space. The decoder h_g then generates output molecules with the desired properties by moving along the direction. In MoleculeSTM, with the pretrained joint representation space, we can accomplish this task by injecting the textual description in a zero-shot manner. As shown in Figure 11 (a, b), we need two phases. The first phase is space alignment, where we train an adaptor module to align the representation space of the generative model to the joint representation space of MoleculeSTM. The second phase is latent optimization, where we directly learn the latent code using two similarity scores as the objective function. Finally, decoding the optimized latent code can lead to the output molecules.

Downstream: molecular property prediction. For modeling, we take the pretrained encoder f_c and add a prediction head h_c to predict a categorical-valued or scalar-valued molecular property such as binding affinity or toxicity. Both f_c and h_c are optimized to fit the target property, *i.e.*, in a fine-tuning manner [99, 153].

2.2. Two Principles for Downstream Task Design

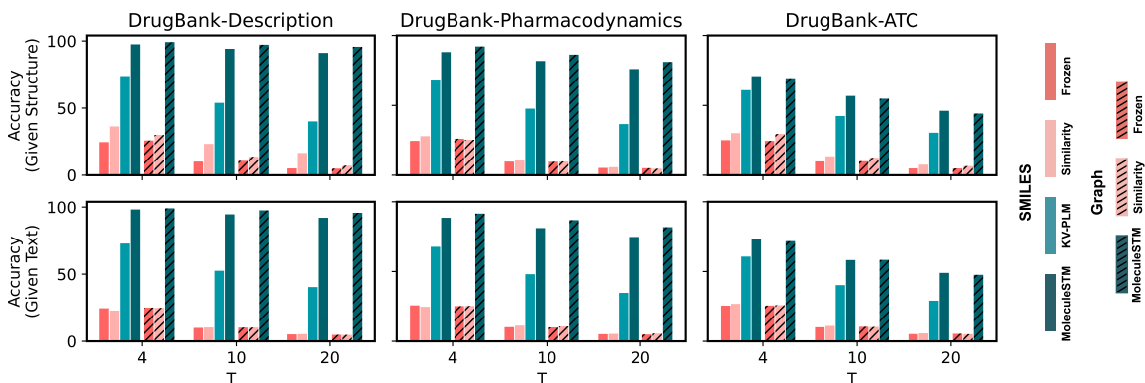
We want to emphasize that for these downstream tasks, the language model in the pretrained MoleculeSTM reveals certain appealing attributes for molecule modeling and drug discovery. We summarize the two key points below.

Open vocabulary. Language is by nature open vocabulary and free form [76]. The large language model has proven its generalization ability in various art-related applications [177, 191, 193], and we find that it can also provide promising and insightful observations for drug discovery tasks. In this vein, our method is not limited to a fixed set of

pre-defined molecule-related annotations but can support the exploration of novel biochemical concepts with unbound vocabulary. One example is the drug re-purposing. Suppose we have a textual description for a new disease or protein target functionality. In that case, we can obtain its similarity with all the existing drugs using MoleculeSTM and retrieve the drugs with the highest rankings, which can be adopted for the later stages, such as clinical trials. Another example is text-based lead optimization. We use natural language to depict an entirely new property, which can be reflected in the generated molecules after the optimization.

Compositionality. Another attribute is compositionality. In natural language, a complex concept can be expressed by decomposing it into simple concepts. This is crucial for certain domain-specific tasks, *e.g.*, multi-objective lead optimization [103] where we need

(a) Structure-text Retrieval Results



(b) Drug Re-purposing Cases (ATC)

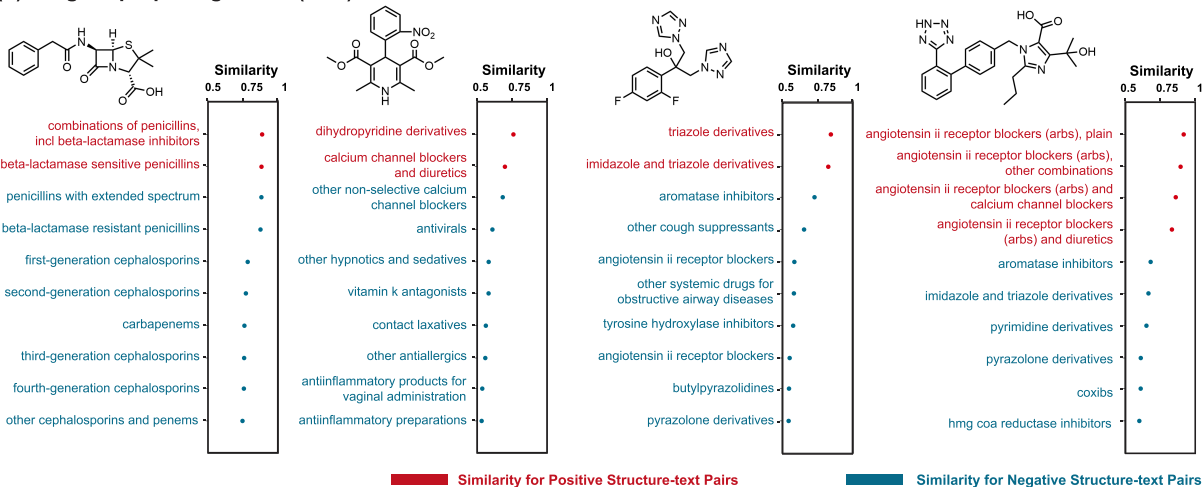


Figure 10. (a) Accuracy for the zero-shot structure-text retrieval downstream tasks. The three datasets (DrugBank-Description, DrugBank-Pharmacodynamics, and DrugBank-ATC) are extracted from DrugBank, and $T \in \{4, 10, 20\}$. Furthermore, two settings are considered: given a chemical structure to retrieve the most similar description and given a textual description to retrieve the most similar chemical structure. (b) Four case studies on ATC retrieval. The chemical structure is given (shown in the molecular graph accordingly) and the goal is to retrieve the ATC labels with the highest similarities. For visualization, four chemical structures are displayed, along with 10 (out of 600) most similar ATC labels, where red and blue colors mark the positive and negative labels, respectively.

to generate molecules with multiple desired properties simultaneously. Existing solutions are either (1) learning one classifier for each desired property and doing filtering on a large candidate pool [113] or (2) optimizing a retrieval database to modify molecules to achieve the multi-objective goal [257]. The main limitation is that the success ratio highly depends on the availability of the labeled data for training the classifier or the retrieval database. While with the language model in MoleculeSTM, we provide an alternative solution. We first craft a natural text, called the text prompt, as the task description. The text prompt can be multi-objective and consists of the description for each property (*e.g.*, “molecule is soluble in water and has high permeability”). With the pretrained joint space between chemical structures and textual descriptions, MoleculeSTM can transform the molecule property compositionality problem into the language compositionality problem, which is more tractable using the language model.

2.3. Downstream: Zero-shot Structure-text Retrieval

Experiments. For the zero-shot retrieval, we construct three datasets from DrugBank [261]. DrugBank is by far the most comprehensive database for drug-like molecules. Here we extract three fields in DrugBank: the description field, the pharmacodynamics field, and the anatomical therapeutic chemical (ATC) field. These fields illustrate the chemical properties and drug effects on the target organism. Then the retrieval task can be viewed as a T -choose-one multiple-choice problem, where T is the number of choices. Specifically, we have two settings: (1) given chemical structure to retrieve the textual description and (2) given the textual description to retrieve the chemical structure. The retrieval accuracy is used as the evaluation metric.

Baselines. We first consider two baselines with the pretrained single-modal encoders [13, 106, 153]. (1) *Frozen* is that we take the pretrained encoders for the two branches and two randomly initialized projectors. (2) *Similarity* is that we take the similarity from a single branch only. For example, in the first setting, when given chemical structure, we retrieve the most similar chemical structure from PubChemSTM, then we take the corresponding paired text representation in PubChemSTM as the proxy representation. Based on this, we can calculate the similarity score between the proxy representation and T requested text representations. (3) We further consider the third baseline *KV-PLM* [280], a pretrained multi-modal model on SMILES-text pairs.

Results. The zero-shot retrieval results are shown in Figure 10 (a). First, we observe that all the algorithms’ accuracies are quite similar between the two settings. Then, as expected, we observe that the baseline *Frozen* performs no better than the random guess because of the randomly-initialized projectors. The *Similarity* baseline is better than the chance performance by a modest margin, verifying that the pretrained single-modality does

learn semantic information but cannot generalize well between modalities. KV-PLM, on the other hand, learns semantically meaningful information from SMILES-text pairs, and thus it achieves much higher accuracies on three datasets. For MoleculeSTM, the graph representation from GNNs has higher accuracy on Description and Pharmacodynamics than the SMILES representation from the transformer model; yet, both of them outperform all the other methods on three datasets and two settings by a large margin. For example, the accuracy improvements are around 50%, 40%, and 15% compared to the best baseline with $T = 20$. Such large improvement gaps verify that MoleculeSTM can play a better role in understanding and bridging the two modalities of molecules.

Case study on drug re-purposing analysis. In Figure 10 (b), we further show four case studies on the retrieval quality of ATC. In specific, given the molecule’s chemical structure, we take 10 (out of 600) most similar ATC labels. It is observed that MoleculeSTM is able to retrieve the ground-truth ATC labels with high rankings.

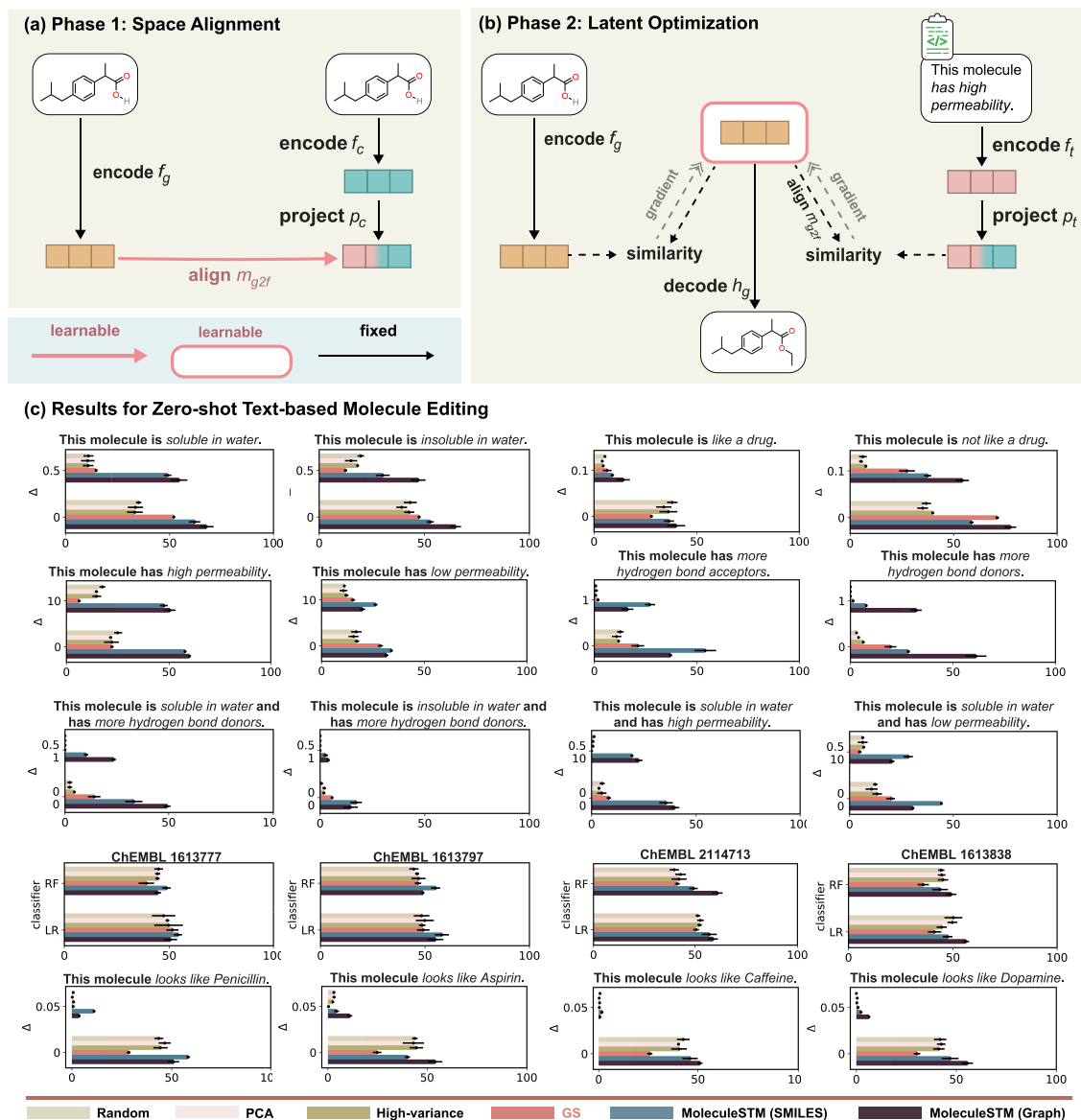


Figure 11. Pipelines (a, b) and results (c) for the zero-shot text-based molecule editing. Text-based molecule editing includes two phases: (a) space alignment and (b) latent optimization. In phase 1, for each chemical structure, the latent representation from a pretrained generative model is mapped to the representation from the pretrained MoleculeSTM. In phase 2, a latent representation is optimized to balance its similarities with representations of the input molecule and a text prompt, and the optimization is done by gradient descent. Finally, the optimized latent code is utilized to generate molecules using the decoder from the generative model, h_g . (c) Satisfactory hit ratios of four types text-based editing tasks. For eight single-objective, four multi-objective, and four drug relevance editing tasks, two satisfactory thresholds (Δ) are considered (see the y -axis in the respective figures). For four ChEMBL binding-affinity-based editing tasks (detailed text prompts are in the supplementary material), one satisfactory threshold and two classifiers (RF for random forest and LR for logistic regression) are considered.

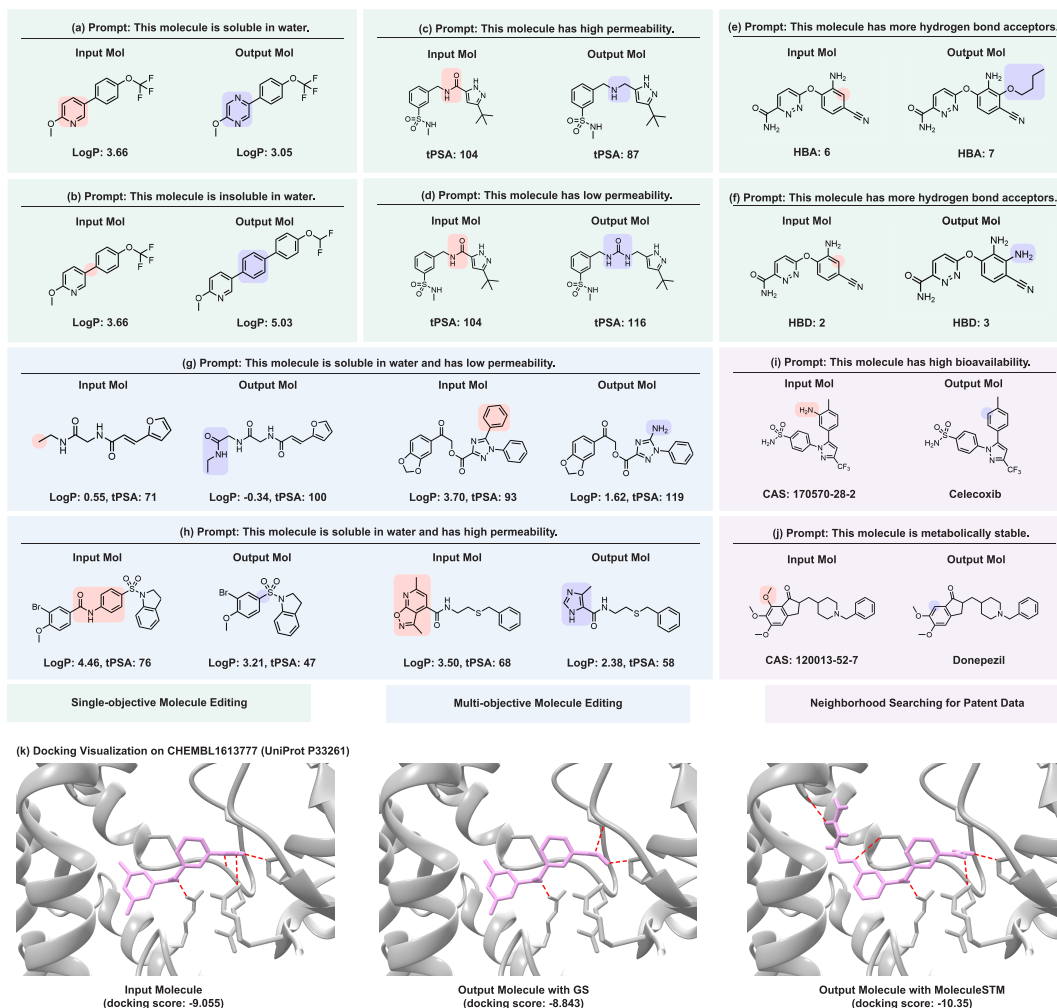


Figure 12. Visual analysis on text-based molecule editing. **Single-objective editing.** Addition, removal, and replacement of functional groups or cores of the molecules are common ways to alter molecular properties, such as water solubility (measured by LogP), permeability (measured by tPSA), and the number of hydrogen bond acceptors/donors. (a) Switching a pyridine to a pyrimidine increases water solubility. (b) Inserting a benzene linkage decreases water solubility. (c) Changing an amide to an amine improves permeability. (d) Replacing an amide with a urea reduces permeability. (e) Adding a butyl ether increases the number of hydrogen bond acceptors. (f) Attaching an amino group contributes to more hydrogen bond donors. **Multi-objective editing.** The solubility and permeability of a molecule can change in the same or opposite directions. (g) Adding polar groups, *e.g.*, amide and amine, and removing hydrophobic components, *e.g.*, methyl and phenyl, are consistent with greater solubility and lower permeability. (h) Both examples generate more soluble and permeable molecules, such as removing a polar amide together with a lipophilic benzene (left), and replacing a [1,2]oxazolo[5,4-*b*]pyridine substituent with an imidazole (right). **Case studies using patented drug analogs.** Text prompts for better drug-like properties are able to generate approved drugs from their analogs. (i) Removing the amino group from the analog improves its intestinal permeability, consistent with higher bioavailability. This edit generates Celecoxib. (j) Replacing the trimethoxy benzene in the analog to dimethoxy benzene increases its metabolic stability, which yields Donepezil. **Docking visualization for binding-affinity-based editing.** The text prompt is from ChEMBL 1613777 (“This molecule is tested positive in an assay that are inhibitors and substrates of an enzyme protein. It uses molecular oxygen inserting one oxygen atom into a substrate, and reducing the second into a water molecule.”). We visualize the results of input molecule and output molecules with GS and MoleculeSTM. The receptors and ligands are marked in grey and plum, respectively, and the hydrogen bonds are marked in red dashed lines. The GS edits the benzene into a pyridine but keeps the rest of the molecule intact, which results in an indistinguishable docking score. However, MoleculeSTM extends the input scaffold along the binding pocket and forms additional hydrogen bonds in a wider range with the receptor and can better stabilize the complex.

2.4. Downstream: Zero-shot Text-based Molecule Editing

Experiments. For molecule editing, we randomly sample 200 molecules from ZINC [105] and a text prompt as the inputs. Four categories of text prompts have been covered: (1) *Single-objective editing* is the text prompt using the single drug-related property for editing, such as “molecule with high solubility” and “molecule more like a drug”. (2) *Multi-objective (compositionality) editing* is the text prompt applying multiple properties simultaneously, such as “molecule with high solubility and high permeability”. (3) *Binding-affinity-based editing* is the text prompt for assay description, where each assay corresponds to one binding affinity task. A concrete example is ChEMBL 1613777 [168] with prompt as “This molecule is tested positive in an assay that are inhibitors and substrates of an enzyme protein. It uses molecular oxygen inserting one oxygen atom into a substrate, and reducing the second into a water molecule.”. The output molecules should possess higher binding affinity scores. (4) *Drug relevance editing* is the text prompt to make molecules structurally similar to certain common drugs, e.g., “this molecule looks like Penicillin”. We expect the output molecules to be more similar to the target drug than the input drug. For more detailed descriptions of the text prompts, please check the supplementary information. The evaluation of each category of text prompt is the satisfactory hit ratio, as discussed in the Methods Section.

Baselines. We consider four baselines. The first three baselines [152] modify the representation of input molecules, followed by the decoding to the molecule space. *Random* is that we take a random noise as the perturbation to the representation of input molecules. *PCA* is that we take the eigenvectors as latent directions, where the eigenvectors are obtained after decomposing the latent representation of input molecules using principle component analysis (PCA). *High Variance* is that we take the latent representation dimension with the highest variance and apply the one-hot encoding on it as a semantic direction for editing. In addition, we also consider a baseline directly modifying the molecule space, the *genetic search (GS)*. It is a variant of graph genetic algorithm [109], while the difference is that GS does a random search instead of a guided search by a reward function since no retrieval database is available in the zero-shot setting.

Results. First, we provide the quantitative results for 20 editing tasks across four editing task types in Figure 11. The empirical results illustrate that the satisfactory hit ratios of MoleculeSTM are the best among all 20 tasks. It verifies that, for both SMILES and molecular graph encoders, MoleculeSTM enables a better semantic understanding of the natural language to explore output molecules with the desired properties. Next, we scrutinize the quality of output molecules in Figure 12 with detailed analysis as follows.

Visual analysis on single-objective molecule editing. We visually analyze the difference between input and output molecules using the single-objective property. Typical modifications are the addition, removal, and replacement of functional groups or cores of the

molecules. For example, Figure 12 (a) and (b) show two different edits on the same molecule leading to opposite directions in solubility change depending on the text prompt. Replacement of pyridine to a pyrazine core improves the solubility, while insertion of a benzene linkage yields an insoluble molecule. In Figure 12 (c) and (d), changing an amide linkage to an alkyl amine and an urea results in higher and lower permeability of the edited molecules, respectively. Finally, Figure 12 (e) and (f) add a butyl ether and a primary amine to the exact position of the molecule, bringing more hydrogen bond acceptors and donors, respectively.

Visual analysis on multi-objective molecule editing. We further analyze the multi-objective (compositional) property editing. Water solubility improvement and permeability reduction are consistent when introducing polar groups to the molecule and removing lipophilic hydrocarbons, such as an amide or primary amine replacing a methyl or phenyl in Figure 12 (g). However, higher solubility and permeability are achievable if polar functionalities are removed or reduced in number together with hydrophobic components. For example, in Figure 12 (h), an amide and a benzene linkage are both removed in the left case, and a *[1,2]oxazolo[5,4-b]pyridine* substituent is replaced by a water-soluble imidazole with a smaller polar surface in the right case.

Case studies on neighborhood searching for patent drug molecules. In drug discovery, improvement of drug-like properties of lead molecules is crucial for finding drug candidates [103]. Herein we demonstrate two examples of generating approved drugs from their patented analogs by addressing their property deficiencies based on text prompts. Figure 12 (i) generates Celecoxib from its amino-substituted derivative [232], where the removal of the amino group yields a greater intestinal permeability of the molecule leading to higher bioavailability [39]. In Figure 12 (j), the trimethoxy benzene moiety, an electron-rich arene known to undergo oxidative phase I metabolisms [78], is replaced by a dimethoxy arene in Donepezil by calling for a metabolically stable molecule.

In summary, we conduct rich experiments on four types and 20 text-based molecule editing tasks, where the satisfactory hit ratios of MoleculeSTM are superior to baseline methods. Moreover, our editing results can match the expected outcomes based on chemistry domain knowledge. Both quantitative and qualitative results illustrate that MoleculeSTM can learn semantically meaningful information useful for domain applications, which encourages us to explore more challenging tasks with MoleculeSTM in the future.

2.5. Downstream: Molecular Property Prediction

Experiments. One advantage for MoleculeSTM is that the pretrained chemical structure representation shares information with the external domain knowledge, and such implicit bias can be beneficial for the property prediction tasks. Similar to previous works on molecule pretraining [99, 153], we adopt the MoleculeNet benchmark [263]. It

contains eight single-modal binary classification datasets to evaluate the expressiveness of the pretrained molecule representation methods. The evaluation metric is the area under the receiver operating characteristic curve (ROC-AUC) [21].

Baselines. We consider two types of chemical structures, the SMILES string and the molecular graph. For the SMILES string, we take three baselines: the *randomly initialized* models and two pretrained language models (*MegaMolBART* [106] and *KV-PLM* [280]). For the molecular graph, in addition to the *random initialization*, we consider five pretraining-based methods as baselines: *AttrMasking* [99], *ContextPred* [99], *InfoGraph* [227], *MolCLR* [256], and *GraphMVP* [142].

Results. As shown in Table 18, we first observe that pretraining-based methods improve the overall classification accuracy compared to the randomly-initialized ones. MoleculeSTM on the SMILES string has consistent improvements on six out of eight tasks compared to the three baselines. MoleculeSTM on the molecular graph performs the best on four out of eight tasks, while it performs comparably to the best baselines in other four tasks. In both cases, the overall performances (*i.e.*, taking an average across all eight tasks) of MoleculeSTM are the best among all the methods.

3. Discussion

In this work, we have presented a multi-modal model, MoleculeSTM, to illustrate the effectiveness of incorporating textual descriptions for molecule representation learning. On two newly proposed zero-shot tasks and one standard property prediction benchmark, we confirmed consistently improved performance of MoleculeSTM compared to the existing methods. Additionally, we observed that MoleculeSTM can retrieve novel drug-target relations and successfully modify molecule substructures to gain the desired properties. These functionalities may accelerate various downstream drug discovery practices, such as

Table 18. Downstream results on eight binary classification datasets from MoleculeNet. The randomly initialized baselines are marked in "-". For other baselines on the SMILES string, MegaMolBART is pretrained on 500M molecules from ZINC, and KV-PLM is pretrained on 10K structure-text pairs from PubChem. For other baselines on the molecular graph, we have five pretraining baselines pretrained on 50K molecules from GEOM. Meanwhile, MoleculeSTM takes the pretrained MegaMolBART and GraphMVP on SMILES and graph, respectively, and continues training on PubChemSTM dataset. We use ROC-AUC for evaluation, and the best results are marked in **bold**.

	method	BBBP \uparrow	Tox21 \uparrow	ToxCast \uparrow	Sider \uparrow	ClinTox \uparrow	MUV \uparrow	HIV \uparrow	Bace \uparrow	Avg \uparrow
SMILES	-	66.54 \pm 0.95	71.18 \pm 0.67	61.16 \pm 1.15	58.31 \pm 0.78	88.11 \pm 0.70	62.74 \pm 1.57	70.32 \pm 1.51	80.02 \pm 1.66	69.80
	MegaMolBART	68.89 \pm 0.17	73.89 \pm 0.67	63.32 \pm 0.79	59.52 \pm 1.79	78.12 \pm 4.62	61.51 \pm 2.75	71.04 \pm 1.70	82.46\pm0.84	69.84
	KV-PLM	70.50 \pm 0.54	72.12 \pm 1.02	55.03 \pm 1.65	59.83 \pm 0.56	89.17\pm2.73	54.63 \pm 4.81	65.40 \pm 1.69	78.50 \pm 2.73	68.15
	MoleculeSTM	70.75\pm1.90	75.71\pm0.89	65.17\pm0.37	63.70\pm0.81	86.60 \pm 2.28	65.69\pm1.46	77.02\pm0.44	81.99 \pm 0.41	73.33
Graph	-	63.90 \pm 2.25	75.06 \pm 0.24	64.64 \pm 0.76	56.63 \pm 2.26	79.86 \pm 7.23	70.43 \pm 1.83	76.23 \pm 0.80	73.14 \pm 5.28	69.99
	AttrMask	67.79 \pm 2.60	75.00 \pm 0.20	63.57 \pm 0.81	58.05 \pm 1.17	75.44 \pm 8.75	73.76 \pm 1.22	75.44 \pm 0.45	80.28 \pm 0.04	71.17
	ContextPred	63.13 \pm 3.48	74.29 \pm 0.23	61.58 \pm 0.50	60.26 \pm 0.77	80.34 \pm 3.79	71.36 \pm 1.44	70.67 \pm 3.56	78.75 \pm 0.35	70.05
	InfoGraph	64.84 \pm 0.55	76.24 \pm 0.37	62.68 \pm 0.65	59.15 \pm 0.63	76.51 \pm 7.83	72.97 \pm 3.61	70.20 \pm 2.41	77.64 \pm 2.04	70.03
	MolCLR	67.79 \pm 0.52	75.55 \pm 0.43	64.58 \pm 0.07	58.66 \pm 0.12	84.22 \pm 1.47	72.76 \pm 0.73	75.88 \pm 0.24	71.14 \pm 1.21	71.32
	GraphMVP	68.11 \pm 1.36	77.06\pm0.35	65.11\pm0.27	60.64 \pm 0.13	84.46 \pm 3.10	74.38\pm2.00	77.74\pm2.51	80.48 \pm 2.68	73.50
	MoleculeSTM	69.98\pm0.52	76.91 \pm 0.51	65.05 \pm 0.39	60.96\pm1.05	92.53\pm1.07	73.40 \pm 2.90	76.93 \pm 1.84	80.77\pm1.34	74.57

re-purposing and multi-objective lead optimization. Furthermore, the outcomes of such downstream tasks have been found to be consistent with the feedback from chemistry experts, reflecting the domain knowledge exploration ability of MoleculeSTM.

One limitation of this work is data insufficiency. Although PubChemSTM is $28\times$ larger than the dataset used in existing works, it can be further improved and may require support from the entire community in the future. The second bottleneck of this work is the expressiveness of chemical structure models, including the SMILES encoder, the GNN encoder, and the SMILES-based molecule generative model. The development of more expressive architectures is perpendicular to this work and can be feasibly adapted to our multi-modal pretraining framework.

For future directions, we would like to extend MoleculeSTM from cheminformatics (small molecules) to bioinformatics tasks (proteins and genomics), which have richer textual information. This also enables us to consider structure-based drug design problems such as protein-ligand binding affinity and fragment design. Besides, the 3D geometric information has become more important for small molecules and polymers and can thus be merged into our foundation model. Last but not least, the joint space between chemical structure and text learned in this work can be further utilized for under-explored problems in AI for drug discovery, including but not limited to out-of-distribution prediction, few-shot prediction, multi-task learning, etc.

4. Methods

This section provides brief descriptions of certain modules in both pretraining and downstream tasks. Detailed specifications, such as dataset construction, model architectures, and hyperparameters, can be found in the supplementary information.

4.1. MoleculeSTM Pretraining

Dataset construction. For the structure-text pretraining, we consider the PubChem database [121] as the data source. PubChem includes 112M molecules, which is one of the largest public databases for molecules. The PubChem database has many fields, and previous work [280] uses the synonym field to match with an academic paper corpus [162], resulting in a dataset with 10K structure-text pairs. Meanwhile, the PubChem database has another field called "string" with more comprehensive and versatile molecule annotations. We utilize this field to construct a large-scale dataset called PubChemSTM, consisting of 250K molecules and 281K structure-text pairs.

In addition, even though PubChemSTM is the largest dataset with textual descriptions, its dataset size is comparatively small compared to the peers from other domains (*e.g.*, 400M in the vision-language domain [191]). To mitigate such a data insufficiency issue, we

adopt the pretrained models from existing checkpoints and then conduct the end-to-end pretraining, as will be discussed next.

Chemical structure branch f_c . This work considers two types of chemical structures: the SMILES string views the molecule as a sequence and the 2D molecular graph takes the atoms and bonds as the nodes and edges, respectively. Then based on the chemical structures, we apply a deep learning encoder f_c to get a latent vector as molecule representation. Specifically, for the SMILES string, we take the encoder from MegaMolBART [106], which is pretrained on 500M molecules from ZINC database [224]. For the molecular graph, we take a pretrained graph isomorphism network (GIN) [266] using GraphMVP pretraining [153]. GraphMVP is doing a multi-view pretraining between the 2D topologies and 3D geometries on 250K conformations from GEOM dataset [9]. Thus, though we are not explicitly utilizing the 3D geometries, the state-of-the-art pretrained GIN models can implicitly encode such information.

Textual description branch f_t . The textual description branch provides a high-level description of the molecule’s functionality. We can view this branch as domain knowledge to strengthen the molecule representation. Such domain knowledge is in the form of natural language, and we use the BERT model [45] as the text encoder f_t . We further adapt the pretrained SciBERT [13], which was pretrained on the textual data from the chemical and biological domain.

Contrastive pretraining. For the MoleculeSTM pretraining, we adopt the contrastive learning strategy, *e.g.*, EBM-NCE [153] and InfoNCE [181]. EBM-NCE and InfoNCE align the structure-text pairs for the same molecule and contrast the pairs for different molecules simultaneously. We consider the selection of contrastive pretraining methods as one important hyperparameter. The objective for EBM-NCE and InfoNCE are

$$\begin{aligned} \mathcal{L}_{\text{EBM-NCE}} &= -\frac{1}{2} \left(\mathbb{E}_{\mathbf{x}_c, \mathbf{x}_t} [\log \sigma(E(\mathbf{x}_c, \mathbf{x}_t))] + \mathbb{E}_{\mathbf{x}_c, \mathbf{x}'_t} [\log(1 - \sigma(E(\mathbf{x}_c, \mathbf{x}'_t)))] \right) \\ &\quad + \mathbb{E}_{\mathbf{x}_c, \mathbf{x}_t} [\log \sigma(E(\mathbf{x}_c, \mathbf{x}_t))] + \mathbb{E}_{\mathbf{x}'_c, \mathbf{x}_t} [\log(1 - \sigma(E(\mathbf{x}'_c, \mathbf{x}_t)))] \Big), \\ \mathcal{L}_{\text{InfoNCE}} &= -\frac{1}{2} \mathbb{E}_{\mathbf{x}_c, \mathbf{x}_t} \left[\log \frac{\exp(E(\mathbf{x}_c, \mathbf{x}_t))}{\exp(E(\mathbf{x}_c, \mathbf{x}_t)) + \sum_{\mathbf{x}'_t} \exp(E(\mathbf{x}_c, \mathbf{x}'_t))} + \log \frac{\exp(E(\mathbf{x}_c, \mathbf{x}_t))}{\exp(E(\mathbf{x}_c, \mathbf{x}_t)) + \sum_{\mathbf{x}'_c} \exp(E(\mathbf{x}'_c, \mathbf{x}_t))} \right], \end{aligned} \tag{4.1}$$

where \mathbf{x}_c and \mathbf{x}_t form the structure-text pair for each molecule, and \mathbf{x}'_c and \mathbf{x}'_t are the negative samples randomly sampled from the noise distribution, which we use the empirical data distribution. $E(\cdot)$ is the energy function with a flexible formulation, and we use the dot product on the jointly learned space, *i.e.*, $E(\mathbf{x}_c, \mathbf{x}_t) = \langle p_c \circ f_c(\mathbf{x}_c), p_t \circ f_t(\mathbf{x}_t) \rangle$.

4.2. Downstream: Zero-shot Structure-text Retrieval

The retrieval task can be viewed as a multiple-choice problem (T -choose-1), where all the encoders (f_c, f_t) and projectors (p_c, p_t) are pretrained from MoleculeSTM, and stay frozen

in this downstream task. An example for the retrieval task of setting (1) is

$$\text{Retrieval}(\mathbf{x}_c) = \arg \max_{\tilde{\mathbf{x}}_t} \left\{ \langle p_c \circ f_c(\mathbf{x}_c), p_t \circ f_t(\tilde{\mathbf{x}}_t) \rangle \mid \tilde{\mathbf{x}}_t \in \mathbb{T} \text{ textual descriptions} \right\}. \quad (4.2)$$

4.3. Downstream: Zero-shot Text-based Molecule Editing

In the molecule editing task, both the MoleculeSTM (f_c, p_c, f_t, p_t) and a pretrained molecule generative model (f_g, h_g) are frozen. Our editing pipeline can be split into two phases: the space alignment phase and the latent optimization phase.

Phase 1: space alignment. In this phase, the goal is to learn an adaptor module to align the representation space of the generative model to the joint representation space of MoleculeSTM. The objective function is

$$\mathcal{L} = \|m_{g2f} \circ f_g(\mathbf{x}_c) - p_c \circ f_c(\mathbf{x}_c)\|^2, \quad (4.3)$$

where m_{g2f} is the adaptor module optimized to align the two latent spaces.

Phase 2: latent optimization. In this phase, given an input molecule $\mathbf{x}_{c,\text{in}}$ and a text prompt \mathbf{x}_t , the goal is to optimize a latent code w directly. The optimal w should be close to the representations of $\mathbf{x}_{c,\text{in}}$ and \mathbf{x}_t simultaneously, as:

$$w = \arg \min_{w \in \mathcal{W}} \left(\mathcal{L}_{\text{cosine-sim}}(m_{g2f}(w), p_t \circ f_t(\mathbf{x}_t)) + \lambda \cdot \mathcal{L}_{l_2}(w, f_g(\mathbf{x}_{c,\text{in}})) \right), \quad (4.4)$$

where $\mathcal{L}_{\text{cosine-sim}}$ is the cosine-similarity, and \mathcal{L}_{l_2} is the l_2 distance, and λ is a coefficient to balance these two similarity terms. Finally, after we optimize the latent code w , we will do decoding using the decoder from the pretrained generative model to obtain the output molecule: $\mathbf{x}_{c,\text{out}} = h_g(w)$.

Evaluation. The evaluation metric is the satisfactory hit ratio. Suppose we have an input molecule $\mathbf{x}_{c,\text{in}}$ and a text prompt \mathbf{x}_t , the editing algorithm will generate an output molecule $\mathbf{x}_{c,\text{out}}$. Then we use the hit ratio to measure if the output molecule can satisfy the conditions as indicated in the text prompt.

$$\text{hit}(\mathbf{x}_{c,\text{in}}, \mathbf{x}_t) = \begin{cases} 1, & \exists \lambda, \text{ s.t. } \mathbf{x}_{c,\text{out}} = h_g(w; \lambda) \wedge \text{satisfy}(\mathbf{x}_{c,\text{in}}, \mathbf{x}_{c,\text{out}}, \mathbf{x}_t) \\ 0, & \text{otherwise} \end{cases}, \quad (4.5)$$

$$\text{hit}(t) = \frac{\sum_{i=1}^N \text{hit}(\mathbf{x}_{c,\text{in}}^i, \mathbf{x}_t)}{N},$$

where N is the total number of editing outputs, and $\text{satisfy}(\cdot)$ is the satisfaction condition. It is task-specific, and we list the five key points below. (1) For single-objective property-based editing, we use the logarithm of partition coefficient (LogP), quantitative estimate of drug-likeness (QED), and topological polar surface area (tPSA) as the proxies to measure the molecule solubility [134], drug likeness [17], and permeability [56], respectively. The count of hydrogen bond acceptors (HBA) and hydrogen bond donors (HBD) are calculated explicitly. It will be a successful hit once the measurement difference between the input

molecule and output molecule is above a certain threshold Δ . (2) For multiple-objective property-based editing, we feed in a text prompt describing multiple properties' composition. The Δ is composed of the threshold on each individual property, and a successful hit needs to satisfy all the properties simultaneously. (3) For binding-affinity-based editing, we take the ground-truth data from ChEMBL to train a binary classifier, and test if the output molecules have higher confidence than the input molecules, and Δ is fixed to 0. (4) For drug relevance editing, we use Tanimoto similarity to quantify the structural similarity [25]. It will be a hit if the similarity score between the output molecule and target drug is higher than the similarity between the input molecule and target drug by a threshold Δ . (5) Besides, the choice of satisfactory threshold Δ is also task-specific, and the higher the values are, the stricter the satisfaction condition is. The details of the threshold values can be found in the supplementary information.

Chapter 2

Conclusion

So far, I have discussed five of my recent research works on using AI for molecule discovery with multi-modal knowledge. Specifically, I discuss how to fuse multi-modal information like 2D topology, 3D geometry, knowledge graph, and textual description for molecule representation. Since AI for molecule discovery relates to both the machine learning and scientific domains, I would like to discuss the future directions from the following two aspects.

Future Step 1: Machine Learning-specific Interpretation. There exist several ML problems unsolved for molecule representation along the pretraining research line. For example, many topology-based molecule pretraining methods [228, 250] can fail under specific circumstances (*i.e.*, the **negative transfer issue**). Certain geometry-based pretraining baselines [146] also possess such a negative transfer issue. I want to argue that solving this may require understanding the whole **learning dynamics** of the pretraining process [235]. Further, inspired by my recent work on graph pretraining [151], I observe that the negative transfer issue is also affected by graph neural network (GNN) architecture, and studying the learning dynamics of graph representation can also relate to solving the **oversmoothing and oversquashing** issues in the GNN literature.

Future Step 2: Domain-specific Interpretation. The ML community observes the increased quantitative performance for tasks related to molecule representation. However, the study on understanding such benefits from the domain aspect is still lagging behind, *e.g.*, why using generative SSL can be superior to contrastive SSL for certain downstream tasks [146] and how to qualitatively verify the extra information obtained during geometric pretraining [154]. As an initial work along this direction, I have two foundation model [20] projects to solve this, named MoleculeSTM [149] and ProteinDT [156]. MoleculeSTM aims to bridge the gap between the molecule’s chemical structure and textual annotation. Such two branches are complementary and combining both can bring in benefits such as making the language model to understand the chemical structures, which enables us to accomplish challenging tasks such as text-based molecule retrieval and editing. ProteinDT has a similar

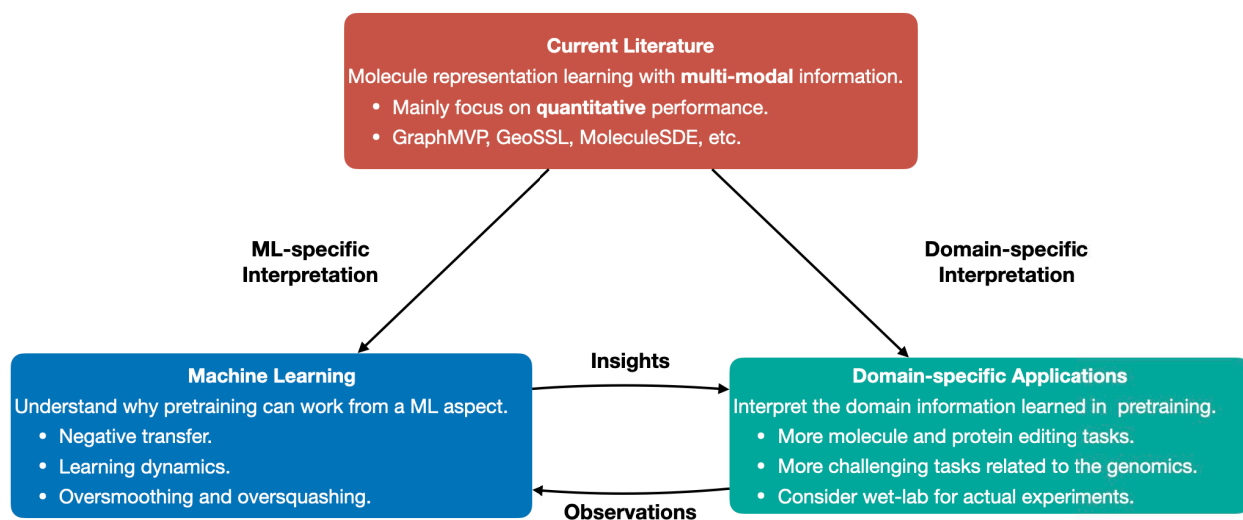


Figure 13. A road map for future direction.

goal, yet the main data structure considered is the protein instead of the small molecule. Meanwhile, both foundation model projects are primary steps along this research line, and I would like to continue exploring **domain-specific editing tasks** and more **challenging tasks related to genomics**. The **ultimate goal** is to design a foundation model for molecule design, which can be further utilized for **wet-lab experiments**. I believe this is a promising direction in applying AI tools for solving scientific tasks, with a real impact on molecule discovery.

Last but not least, I would like to highlight that such two directions are not mutually exclusive but complementary: the ML-specific interpretation can provide more insights into designing the algorithms for domain-specific tasks, and the observations from domain applications can help verify or guide the ML interpretation. I think both directions are worth investigating, and their interpretations can jointly help reach the goal of AI for molecule discovery.

Bibliography

- [1] Saurabh AGGARWAL : Targeted cancer therapies. *Nature reviews. Drug discovery*, 9(6):427, 2010.
- [2] Moayad ALNAMMI, Shengchao LIU, Spencer S ERICKSEN, Gene E ANANIEV, Andrew F VOTER, Song GUO, James L KECK, F Michael HOFFMANN, Scott A WILDMAN et Anthony GITTER : Evaluating scalable supervised learning for synthesizable-on-demand chemical libraries. *ChemRxiv*, 2021.
- [3] Waleed AMMAR, Dirk GROENEVELD, Chandra BHAGAVATULA, Iz BELTAGY, Miles CRAWFORD, Doug DOWNEY, Jason DUNKELBERGER, Ahmed ELGOHARY, Sergey FELDMAN, Vu HA *et al.* : Construction of the literature graph in semantic scholar. *arXiv preprint arXiv:1805.02262*, 2018.
- [4] Michael ARBEL, Liang ZHOU et Arthur GRETTON : Generalized energy based models. *arXiv preprint arXiv:2003.05033*, 2020.
- [5] Sanjeev ARORA, Hrishikesh KHANDEPARKAR, Mikhail KHODAK, Orestis PLEVRAKIS et Nikunj SAUNSHI : A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.
- [6] Kenneth ATZ, Francesca GRISONI et Gisbert SCHNEIDER : Geometric deep learning on molecular representations. *Nature Machine Intelligence*, pages 1–10, 2021.
- [7] Simon AXELROD et Rafael GOMEZ-BOMBARELLI : Geom: Energy-annotated molecular conformations for property prediction and molecular generation. *arXiv preprint arXiv:2006.05531*, 2020.
- [8] Simon AXELROD et Rafael GOMEZ-BOMBARELLI : Molecular machine learning with conformer ensembles. *arXiv preprint arXiv:2012.08452*, 2020.
- [9] Simon AXELROD et Rafael GOMEZ-BOMBARELLI : Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1): 1–14, 2022.
- [10] Philip BACHMAN, R Devon HJELM et William BUCHWALTER : Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

- [11] David BELANGER et Andrew MCCALLUM : Structured prediction energy networks. *In International Conference on Machine Learning*, pages 983–992. PMLR, 2016.
- [12] Mohamed Ishmael BELGHAZI, Aristide BARATIN, Sai RAJESHWAR, Sherjil OZAIR, Yoshua BENGIO, Aaron COURVILLE et Devon HJELM : Mutual information neural estimation. *In International Conference on Machine Learning*, pages 531–540. PMLR, 2018.
- [13] Iz BELTAGY, Kyle LO et Arman COHAN : Scibert: Pretrained language model for scientific text. *In EMNLP*, 2019.
- [14] Yoshua BENGIO, Tristan DELEU, Edward J HU, Salem LAHLOU, Mo TIWARI et Emmanuel BENGIO : Gflownet foundations. *arXiv preprint arXiv:2111.09266*, 2021.
- [15] Yoshua BENGIO, J’erôme LOURADOUR, Ronan COLLOBERT et Jason WESTON : Curriculum learning. *In Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.
- [16] Hangrui BI, Hengyi WANG, Chence SHI, Connor W. COLEY, Jian TANG et Hongyu GUO : Non-autoregressive electron redistribution modeling for reaction prediction. *In ICML*, 2021.
- [17] G Richard BICKERTON, Gaia V PAOLINI, J’er’emy BESNARD, Sorel MURESAN et Andrew L HOPKINS : Quantifying the chemical beauty of drugs. *Nature chemistry*, 4(2):90–98, 2012.
- [18] Avrim BLUM et Tom MITCHELL : Combining labeled and unlabeled data with co-training. *In Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.
- [19] Hans-Joachim B"OHM, Alexander FLOHR et Martin STAHL : Scaffold hopping. *Drug discovery today: Technologies*, 1(3):217–224, 2004.
- [20] Rishi BOMMASANI, Drew A HUDSON, Ehsan ADELI, Russ ALTMAN, Simran ARORA, Sydney von ARX, Michael S BERNSTEIN, Jeannette BOHG, Antoine BOSSELUT, Emma BRUNSKILL *et al.* : On the opportunities and risks of foundation models. *arXiv:2108.07258*, 2021.
- [21] Andrew P BRADLEY : The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.
- [22] Johannes BRANDSTETTER, Rob HESSELINK, Elise van der POL, Erik BEKKERS et Max WELLING : Geometric and physical quantities improve e(3) equivariant message passing. *arXiv preprint arXiv:2110.02905*, 2021.
- [23] Nathan BROWN, Marco FISCATO, Marwin HS SEGLER et Alain C VAUCHER : Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3):1096–1108, 2019.
- [24] Tom B BROWN, Benjamin MANN, Nick RYDER, Melanie SUBBIAH, Jared KAPLAN, Prafulla DHARIWAL, Arvind NEELAKANTAN, Pranav SHYAM, Girish SASTRY,

- Amanda ASKELL *et al.* : Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- [25] Darko BUTINA : Unsupervised data base clustering based on daylight’s fingerprint and tanimoto similarity: A fast and automated way to cluster small and large data sets. *Journal of Chemical Information and Computer Sciences*, 39(4):747–750, 1999.
- [26] Mathilde CARON, Ishan MISRA, Julien MAIRAL, Priya GOYAL, Piotr BOJANOWSKI et Armand JOULIN : Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.
- [27] Jintai CHEN, Biwen LEI, Qingyu SONG, Haochao YING, Danny Z CHEN et Jian WU : A hierarchical graph network for 3d object detection on point clouds. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 392–401, 2020.
- [28] Ting CHEN, Simon KORNBLITH, Mohammad NOROUZI et Geoffrey HINTON : A simple framework for contrastive learning of visual representations. *In International conference on Machine Learning*, pages 1597–1607, 2020.
- [29] Xi CHEN, Yan DUAN, Rein HOUTHOOFT, John SCHULMAN, Ilya SUTSKEVER et Pieter ABBEEL : Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *In Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 2180–2188, 2016.
- [30] Xinlei CHEN et Kaiming HE : Exploring simple siamese representation learning. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.
- [31] Zhao CHEN, Vijay BADRINARAYANAN, Chen-Yu LEE et Andrew RABINOVICH : Grad-norm: Gradient normalization for adaptive loss balancing in deep multitask networks. *In International Conference on Machine Learning*, pages 794–803, 2018.
- [32] Stefan CHMIELA, Alexandre TKATCHENKO, Huziel E SAUCEDA, Igor POLTAVSKY, Kristof T SCH"UTT et Klaus-Robert M"ULLER : Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5):e1603015, 2017.
- [33] John D CHODERA et Frank NO’E : Markov state models of biomolecular conformational dynamics. *Current opinion in structural biology*, 25:135–144, 2014.
- [34] Jonathan CLAYDEN, Nick GREEVES et Stuart WARREN : *Organic chemistry*. Oxford university press, 2012.
- [35] The UniProt CONSORTIUM : UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1):D506–D515, 11 2018.
- [36] Gabriele CORSO, Luca CAVALLERI, Dominique BEAINI, Pietro LIÒ et Petar VELICKOVI’C : Principal neighbourhood aggregation for graph nets. *Advances in Neural Information Processing Systems*, 33:13260–13271, 2020.
- [37] F Albert COTTON : *Chemical applications of group theory*. John Wiley & Sons, 1991.

- [38] George E DAHL, Navdeep JAITLEY et Ruslan SALAKHUTDINOV : Multi-task neural networks for QSAR predictions. *arXiv preprint arXiv:1406.1231*, 2014.
- [39] David DAHLGREN et Hans LENNERNÄS : Intestinal permeability and drug absorption: Predictive experimental, computational and in vivo approaches. *Pharmaceutics*, 11(8), 2019.
- [40] Jifeng DAI, Yang LU et Ying-Nian WU : Generative modeling of convolutional neural networks. *arXiv preprint arXiv:1412.6296*, 2014.
- [41] Mindy I DAVIS, Jeremy P HUNT, Sanna HERRGARD, Pietro CICERI, Lisa M WODICKA, Gabriel PALLARES, Michael HOCKER, Daniel K TREIBER et Patrick P ZARRINKAR : Comprehensive analysis of kinase inhibitor selectivity. *Nature biotechnology*, 29(11):1046–1051, 2011.
- [42] John S DELANEY : Esol: estimating aqueous solubility directly from molecular structure. *Journal of chemical information and computer sciences*, 44(3):1000–1005, 2004.
- [43] Mehmet F DEMIREL, Shengchao LIU, Siddhant GARG et Yingyu LIANG : An analysis of attentive walk-aggregating graph neural networks. *arXiv preprint arXiv:2110.02667*, 2021.
- [44] Mehmet F DEMIREL, Shengchao LIU, Siddhant GARG, Zhenmei SHI et Yingyu LIANG : Attentive walk-aggregating graph neural networks. *TMLR*, 2022.
- [45] Jacob DEVLIN, Ming-Wei CHANG, Kenton LEE et Kristina TOUTANOVA : Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [46] Prafulla DHARIWAL et Alexander NICHOL : Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.
- [47] Laurent DINH, Jascha SOHL-DICKSTEIN et Samy BENGIO : Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.
- [48] Jürgen DREWS : Drug discovery: a historical perspective. *Science*, 287(5460):1960–1964, 2000.
- [49] Weitao DU, He ZHANG, Yuanqi DU, Qi MENG, Wei CHEN, Nanning ZHENG, Bin SHAO et Tie-Yan LIU : Se (3) equivariant graph neural networks with complete local frames. *In International Conference on Machine Learning*, pages 5583–5608. PMLR, 2022.
- [50] Yilun DU, Shuang LI, Joshua TENENBAUM et Igor MORDATCH : Improved contrastive divergence training of energy based models. *arXiv preprint arXiv:2012.01316*, 2020.
- [51] Yuanqi DU, Tianfan FU, Jimeng SUN et Shengchao LIU : Molgensurvey: A systematic survey in machine learning models for molecule design. *arXiv preprint arXiv:2203.14500*, 2022.
- [52] David DUVENAUD, Dougal MACLAURIN, Jorge AGUILERA-IPARRAGUIRRE, Rafael GÓMEZ-BOMBARELLI, Timothy HIRZEL, Al'an ASPURU-GUZIK et Ryan P ADAMS :

- Convolutional networks on graphs for learning molecular fingerprints. *arXiv preprint arXiv:1509.09292*, 2015.
- [53] Thomas ENGEL et Johann GASTEIGER : *Applied chemoinformatics: achievements and future opportunities*. John Wiley & Sons, 2018.
- [54] Dumitru ERHAN, Aaron COURVILLE, Yoshua BENGIO et Pascal VINCENT : Why does unsupervised pre-training help deep learning? *In Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 201–208. JMLR Workshop and Conference Proceedings, 2010.
- [55] Peter ERTL, Eva ALTMANN et Jeffrey M MCKENNA : The most common functional groups in bioactive molecules and how their popularity has evolved over time. *Journal of medicinal chemistry*, 63(15):8408–8418, 2020.
- [56] Peter ERTL, Bernhard ROHDE et Paul SELZER : Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties. *Journal of Medicinal Chemistry*, 43(20):3714–3717, 2000. PMID: 11020286.
- [57] Linxi FAN, Guanzhi WANG, Yunfan JIANG, Ajay MANDLEKAR, Yuncong YANG, Haoyi ZHU, Andrew TANG, De-An HUANG, Yuke ZHU et Anima ANANDKUMAR : Minedojo: Building open-ended embodied agents with internet-scale knowledge. *arXiv preprint arXiv:2206.08853*, 2022.
- [58] Xiaomin FANG, Lihang LIU, Jieqiong LEI, Donglong HE, Shanzhuo ZHANG, Jingbo ZHOU, Fan WANG, Hua WU et Haifeng WANG : Chemrl-gem: Geometry enhanced molecular representation learning for property prediction. *arXiv preprint arXiv:2106.06130*, 2021.
- [59] Yin FANG, Qiang ZHANG, Haihong YANG, Xiang ZHUANG, Shumin DENG, Wen ZHANG, Ming QIN, Zhuo CHEN, Xiaohui FAN et Huaajun CHEN : Molecular contrastive learning with chemical element knowledge graph. *arXiv preprint arXiv:2112.00544*, 2021.
- [60] Elizabeth H FINN, Gianluca PEGORARO, Sigal SHACHAR et Tom MISTELI : Comparative analysis of 2d and 3d distance measurements to study spatial genome organization. *Methods*, 123:47–55, 2017.
- [61] Fabian B FUCHS, Daniel E WORRALL, Volker FISCHER et Max WELLING : Se (3)-transformers: 3d roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020.
- [62] Francisco-Javier GAMO, Laura M SANZ, Jaume VIDAL, Cristina DE COZAR, Emilio ALVAREZ, Jose-Luis LAVANDERA, Dana E VANDERWALL, Darren VS GREEN, Vinod KUMAR, Samiul HASAN *et al.* : Thousands of chemical starting points for antimalarial lead identification. *Nature*, 465(7296):305, 2010.

- [63] Wenhao GAO et Connor W COLEY : The synthesizability of molecules proposed by generative models. *Journal of chemical information and modeling*, 60(12):5714–5723, 2020.
- [64] Siddhant GARG et Yingyu LIANG : Functional regularization for representation learning: A unified theoretical perspective. *arXiv preprint arXiv:2008.02447*, 2020.
- [65] Anna GAULTON, Louisa J BELLIS, A Patricia BENTO, Jon CHAMBERS, Mark DAVIES, Anne HERSEY, Yvonne LIGHT, Shaun MCGLINCHEY, David MICHALOVICH, Bissan AL-LAZIKANI *et al.* : ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic acids research*, 40(D1):D1100–D1107, 2012.
- [66] Kaitlyn M GAYVERT, Neel S MADHUKAR et Olivier ELEMENTO : A data-driven approach to predicting successes and failures of clinical trials. *Cell chemical biology*, 23(10):1294–1301, 2016.
- [67] Mario GEIGER et Tess SMIDT : e3nn: Euclidean neural networks. *arXiv preprint arXiv:2207.09453*, 2022.
- [68] Stuart GEMAN et Donald GEMAN : Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, pages 721–741, 1984.
- [69] Justin GILMER, Samuel S SCHOENHOLZ, Patrick F RILEY, Oriol VINYALS et George E DAHL : Neural message passing for quantum chemistry. *In International conference on machine learning*, pages 1263–1272. PMLR, 2017.
- [70] Jonathan GODWIN, Michael SCHAARSCHMIDT, Alexander L GAUNT, Alvaro SANCHEZ-GONZALEZ, Yulia RUBANOVA, Petar VELICKOVIĆ, James KIRKPATRICK et Peter BATTAGLIA : Simple GNN regularisation for 3d molecular property prediction and beyond. *In International Conference on Learning Representations*, 2022.
- [71] Laurent GOMEZ : Decision making in medicinal chemistry: The power of our intuition. *ACS Medicinal Chemistry Letters*, 9(10):956–958, 2018.
- [72] Rafael GÓMEZ-BOMBARELLI, Jennifer N WEI, David DUVENAUD, José Miguel HERNÁNDEZ-LOBATO, Benjamin SANCHEZ-LENGELING, Dennis SHEBERLA, Jorge AGUILERA-IPARRAGUIRRE, Timothy D HIRZEL, Ryan P ADAMS et Alan ASPURUGUZIK : Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.
- [73] Ian GOODFELLOW, Jean POUGET-ABADIE, Mehdi MIRZA, Bing XU, David WARDEFARLEY, Sherjil OZAIR, Aaron COURVILLE et Yoshua BENGIO : Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [74] Sai Krishna GOTTIPATI, Boris SATTAROV, Sufeng NIU, Yashaswi PATHAK, Haoran WEI, Shengchao LIU, Simon BLACKBURN, Karam THOMAS, Connor COLEY, Jian TANG *et al.* : Learning to navigate the synthetically accessible chemical space using

- reinforcement learning. *In International Conference on Machine Learning*, pages 3668–3679. PMLR, 2020.
- [75] Jean-Bastien GRILL, Florian STRUB, Florent ALTCH’E, Corentin TALLEC, Pierre RICHEMOND, Elena BUCHATSKAYA, Carl DOERSCH, Bernardo AVILA PIRES, Zhaohan GUO, Mohammad GHESHLAGHI AZAR *et al.* : Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.
- [76] Xiuye GU, Tsung-Yi LIN, Weicheng KUO et Yin CUI : Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.
- [77] Yuzhi GUO, Jiaxiang WU, Hehuan MA et Junzhou HUANG : Self-supervised pre-training for protein embeddings using tertiary structures. 2022.
- [78] Gordon GUROFF, Jean RENSON, Sidney UDENFRIEND, John W DALY, Donald M JERINA et Bernhard WITKOP : Hydroxylation-induced migration: The nih shift: Recent experiments reveal an unexpected and general result of enzymatic hydroxylation of aromatic compounds. *Science*, 157(3796):1524–1530, 1967.
- [79] Michael GUTMANN et Aapo HYV"ARINEN : Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. *In Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.
- [80] Johannes HACHMANN, Roberto OLIVARES-AMAYA, Sule ATAHAN-EVRENK, Carlos AMADOR-BEDOLLA, Roel S S’ANCHEZ-CARRERA, Aryeh GOLD-PARKER, Leslie VOGT, Anna M BROCKWAY et Al’an ASPURU-GUZIĆ : The harvard clean energy project: large-scale computational screening and design of organic photovoltaics on the world community grid. *The Journal of Physical Chemistry Letters*, 2(17):2241–2251, 2011.
- [81] Thomas A HALGREN : Merck molecular force field. i. basis, form, scope, parameterization, and performance of mmff94. *Journal of computational chemistry*, 17(5-6):490–519, 1996.
- [82] Will HAMILTON, Zhitao YING et Jure LESKOVEC : Inductive representation learning on large graphs. *In Advances in neural information processing systems*, pages 1024–1034, 2017.
- [83] Kaveh HASSANI et Amir Hosein KHASAHMADI : Contrastive multi-view representation learning on graphs. *In International Conference on Machine Learning*, pages 4116–4126. PMLR, 2020.
- [84] Ryuichiro HATAYA, Hideki NAKAYAMA et Kazuki YOSHIZOE : Graph energy-based model for molecular graph generation. *In Energy Based Models Workshop-ICLR 2021*, 2021.

- [85] Paul CD HAWKINS : Conformation generation: the state of the art. *Journal of Chemical Information and Modeling*, 57(8):1747–1756, 2017.
- [86] Kaiming HE, Xinlei CHEN, Saining XIE, Yanghao LI, Piotr DOLL’AR et Ross GIRSHICK : Masked autoencoders are scalable vision learners. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.
- [87] Kaiming HE, Haoqi FAN, Yuxin WU, Saining XIE et Ross GIRSHICK : Momentum contrast for unsupervised visual representation learning. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.
- [88] Irina HIGGINS, Loic MATTHEY, Arka PAL, Christopher BURGESS, Xavier GLOROT, Matthew BOTVINICK, Shakir MOHAMED et Alexander LERCHNER : beta-vae: Learning basic visual concepts with a constrained variational framework. *In International Conference on Learning Representations*, 2017.
- [89] Geoffrey E HINTON : Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.
- [90] Maya HIROHARA, Yutaka SAITO, Yuki KODA, Kengo SATO et Yasubumi SAKAKIBARA : Convolutional neural network based on SMILES representation of compounds for detecting chemical motif. *BMC bioinformatics*, 19(19):83–94, 2018.
- [91] R Devon HJELM, Alex FEDOROV, Samuel LAVOIE-MARCHILDON, Karan GREWAL, Phil BACHMAN, Adam TRISCHLER et Yoshua BENGIO : Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.
- [92] Jonathan HO, Ajay JAIN et Pieter ABBEEL : Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [93] M HONNIBAL, I MONTANI et S VAN LANDEGHEM : Boyd. *A. spaCy: industrial-strength natural language processing in Python*, 2020.
- [94] Chloe HSU, Robert VERKUIL, Jason LIU, Zeming LIN, Brian HIE, Tom SERCU, Adam LERER et Alexander RIVES : Learning inverse folding from millions of predicted structures. *bioRxiv*, 2022.
- [95] Qianjiang HU, Xiao WANG, Wei HU et Guo-Jun QI : Adco: Adversarial contrast for efficient learning of unsupervised representations from self-trained negative adversaries. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1074–1083, 2021.
- [96] Weihua HU, Matthias FEY, Marinka ZITNIK, Yuxiao DONG, Hongyu REN, Bowen LIU, Michele CATASTA et Jure LESKOVEC : Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

- [97] Weihua HU, Matthias FEY, Marinka ZITNIK, Yuxiao DONG, Hongyu REN, Bowen LIU, Michele CATASTA et Jure LESKOVEC : Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.
- [98] Weihua HU, Bowen LIU, Joseph GOMES, Marinka ZITNIK, Percy LIANG, Vijay PANDE et Jure LESKOVEC : Pre-training graph neural networks. *arXiv preprint arXiv:1905.12265*, 2019.
- [99] Weihua HU, Bowen LIU, Joseph GOMES, Marinka ZITNIK, Percy LIANG, Vijay PANDE et Jure LESKOVEC : Strategies for pre-training graph neural networks. *In International Conference on Learning Representations, ICLR*, 2020.
- [100] Ye HU, Dagmar STUMPFE et Jurgen BAJORATH : Recent advances in scaffold hopping: miniperspective. *Journal of medicinal chemistry*, 60(4):1238–1246, 2017.
- [101] Ziniu HU, Yuxiao DONG, Kuansan WANG, Kai-Wei CHANG et Yizhou SUN : Gpt-gnn: Generative pre-training of graph neural networks. *In ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD*, pages 1857–1867, 2020.
- [102] Chin-Wei HUANG, Jae Hyun LIM et Aaron C COURVILLE : A variational perspective on diffusion-based generative models and score matching. *In M. RANZATO, A. BEYGEZIMER, Y. DAUPHIN, P.S. LIANG et J. Wortman VAUGHAN, éditeurs : Advances in Neural Information Processing Systems*, volume 34, pages 22863–22876. Curran Associates, Inc., 2021.
- [103] James P HUGHES, Stephen REES, S Barrett KALINDJIAN et Karen L PHILPOTT : Principles of early drug discovery. *British journal of pharmacology*, 162(6):1239–1249, 2011.
- [104] Aapo HYV"ARINEN et Peter DAYAN : Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.
- [105] John J IRWIN, Teague STERLING, Michael M MYSINGER, Erin S BOLSTAD et Ryan G COLEMAN : Zinc: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7):1757–1768, 2012.
- [106] Ross IRWIN, Spyridon DIMITRIADIS, Jiazhen HE et Esben Jannik BJERRUM : Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022, 2022.
- [107] Abhinav JAIN, Minali UPRETI et Preethi JYOTHI : Improved accented speech recognition using accent embeddings and multi-task learning. *In Interspeech*, pages 2454–2458, 2018.
- [108] Madura KP JAYATUNGA, Wen XIE, Ludwig RUDER, Ulrik SCHULZE et Christoph MEIER : Ai in small-molecule drug discovery: A coming wave. *Nat. Rev. Drug Discov.*, 21:175–176, 2022.
- [109] Jan H JENSEN : A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical science*, 10(12):3567–3572,

- 2019.
- [110] Yuanfeng JI, Lu ZHANG, Jiaxiang WU, Bingzhe WU, Long-Kai HUANG, Tingyang XU, Yu RONG, Lanqing LI, Jie REN, Ding XUE *et al.* : Drugood: Out-of-distribution (ood) dataset curator and benchmark for ai-aided drug discovery—a focus on affinity prediction problems with noise annotations. *arXiv preprint arXiv:2201.09637*, 2022.
 - [111] Dejun JIANG, Zhenxing WU, Chang-Yu HSIEH, Guangyong CHEN, Ben LIAO, Zhe WANG, Chao SHEN, Dongsheng CAO, Jian WU et Tingjun HOU : Could graph neural networks learn better molecular representation for drug discovery? a comparison study of descriptor-based and graph-based models. *Journal of cheminformatics*, 13(1):1–23, 2021.
 - [112] Rui JIAO, Jiaqi HAN, Wenbing HUANG, Yu RONG et Yang LIU : 3d equivariant molecular graph pretraining. *arXiv preprint arXiv:2207.08824*, 2022.
 - [113] Wengong JIN, Regina BARZILAY et Tommi JAAKKOLA : Hierarchical generation of molecular graphs using structural motifs. *In International conference on machine learning*, pages 4839–4848. PMLR, 2020.
 - [114] Jaehyeong JO, Seul LEE et Sung Ju HWANG : Score-based generative modeling of graphs via the system of stochastic differential equations. *arXiv preprint arXiv:2202.02514*, 2022.
 - [115] John JUMPER, Richard EVANS, Alexander PRITZEL, Tim GREEN, Michael FIGURNOV, Olaf RONNEBERGER, Kathryn TUNYASUVUNAKOOL, Russ BATES, Augustin Z’IDEK, Anna POTAPENKO *et al.* : Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.
 - [116] Tero KARRAS, Miika AITTALA, Timo AILA et Samuli LAINE : Elucidating the design space of diffusion-based generative models. *arXiv preprint arXiv:2206.00364*, 2022.
 - [117] Tero KARRAS, Samuli LAINE et Timo AILA : A style-based generator architecture for generative adversarial networks. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.
 - [118] Ramandeep KAUR, Fabio POSSANZA, Francesca LIMOSANI, Stefan BAUROTH, Robertino ZANONI, Timothy CLARK, Giorgio ARRIGONI, Pietro TAGLIATESTA et Dirk M GULDI : Understanding and controlling short-and long-range electron/charge-transfer processes in electron donor–acceptor conjugates. *Journal of the American Chemical Society*, 142(17):7898–7911, 2020.
 - [119] Alex KENDALL, Yarín GAL et Roberto CIPOLLA : Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491, 2018.
 - [120] Prannay KHOSLA, Piotr TETERWAK, Chen WANG, Aaron SARNA, Yonglong TIAN, Phillip ISOLA, Aaron MASCHINOT, Ce LIU et Dilip KRISHNAN : Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

- [121] Sunghwan KIM, Jie CHEN, Tiejun CHENG, Asta GINDULYTE, Jia HE, Siqian HE, Qingliang LI, Benjamin A SHOEMAKER, Paul A THIESSEN, Bo YU *et al.* : Pubchem in 2021: new data content and improved web interfaces. *Nucleic acids research*, 49(D1): D1388–D1395, 2021.
- [122] Diederik P KINGMA et Prafulla DHARIWAL : Glow: generative flow with invertible *limes* 1 convolutions. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 10236–10245, 2018.
- [123] Diederik P KINGMA et Max WELLING : Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [124] Thomas N KIPF et Max WELLING : Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.
- [125] Johannes KLICPERA, Florian BECKER et Stephan G"UNNEMANN : Gemnet: Universal directional graph neural networks for molecules. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.
- [126] Johannes KLICPERA, Shankari GIRI, Johannes T MARGRAF et Stephan G"UNNEMANN : Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv preprint arXiv:2011.14115*, 2020.
- [127] Mario KRENN, Florian H"ASE, AkshatKumar NIGAM, Pascal FRIEDERICH et Alan ASPURU-GUZIK : Self-referencing embedded strings (selfies): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 1(4):045024, 2020.
- [128] Michael KUHN, Ivica LETUNIC, Lars Juhl JENSEN et Peer BORK : The sider database of drugs and side effects. *Nucleic acids research*, 44(D1):D1075–D1079, 2015.
- [129] Greg LANDRUM *et al.* : RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling, 2013.
- [130] Hugo LAROCHELLE, Dumitru ERHAN et Yoshua BENGIO : Zero-data learning of new tasks. In *AAAI*, volume 1, page 3, 2008.
- [131] Gustav LARSSON, Michael MAIRE et Gregory SHAKHNAROVICH : Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593, 2016.
- [132] Yann LECUN, Sumit CHOPRA, Raia HADSELL, M RANZATO et F HUANG : A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.
- [133] Kyoungyeul LEE et Dongsup KIM : In-silico molecular binding prediction for human drug targets using deep neural multi-task learning. *Genes*, 10(11):906, 2019.
- [134] Albert LEO, Corwin HANSCH et David ELKINS : Partition coefficients and their uses. *Chemical Reviews*, 71(6):525–616, 1971.
- [135] Shuang LI, Xavier PUIG, Yilun DU, Clinton WANG, Ekin AKYUREK, Antonio TORRALBA, Jacob ANDREAS et Igor MORDATCH : Pre-trained language models for interactive decision-making. *arXiv preprint arXiv:2202.01771*, 2022.

- [136] Meng LIU, Keqiang YAN, Bora OZTEKIN et Shuiwang JI : Graphebm: Molecular graph generation with energy-based models. *arXiv preprint arXiv:2102.00546*, 2021.
- [137] Shengchao LIU : Exploration on deep drug discovery: Representation and learning. *Master’s Thesis*, TR1854, 2018.
- [138] Shengchao LIU, Moayad ALNAMMI, Spencer S ERICKSEN, Andrew F VOTER, Gene E ANANIEV, James L KECK, F Michael HOFFMANN, Scott A WILDMAN et Anthony GITTER : Practical model selection for prospective virtual screening. *Journal of chemical information and modeling*, 59(1):282–293, 2018.
- [139] Shengchao LIU, Andreea DEAC, Zhaocheng ZHU et Jian TANG : Structured multi-view representations for drug combinations. *In NeurIPS 2020 ML for Molecules Workshop*, 2020.
- [140] Shengchao LIU, Mehmet F DEMIREL et Yingyu LIANG : N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. *Advances in neural information processing systems*, 32, 2019.
- [141] Shengchao LIU, Mehmet F DEMIREL et Yingyu LIANG : N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. *In H. WALLACH, H. LAROCHELLE, A. BEYGELZIMER, F. d extquotesingle ALCH’E-BUC, E. FOX et R. GARNETT, éditeurs : Advances in Neural Information Processing Systems 32*, pages 8464–8476. Curran Associates, Inc., 2019.
- [142] Shengchao LIU, Mehmet Furkan DEMIREL et Yingyu LIANG : N-gram graph: Simple unsupervised representation for graphs, with applications to molecules. *arXiv preprint arXiv:1806.09206*, 2018.
- [143] Shengchao LIU, Weitao DU, Zhiming MA, Hongyu GUO et Jian TANG : A group symmetric stochastic differential equation model for molecule multi-modal pretraining. *In Submission to ICML*, 2023.
- [144] Shengchao LIU, Weitao DU et Jian TANG : Molecule geometric representation learning benchmark. *International Conference on Machine Learning*, 2023.
- [145] Shengchao LIU, Hongyu GUO et Jian TANG : Molecular geometry pretraining with se (3)-invariant denoising distance matching. *arXiv preprint arXiv:2206.13602*, 2022.
- [146] Shengchao LIU, Hongyu GUO et Jian TANG : Molecular geometry pretraining with SE(3)-invariant denoising distance matching. *In ICLR*, 2023.
- [147] Shengchao LIU, Yingyu LIANG et Anthony GITTER : Loss-balanced task weighting to reduce negative transfer in multi-task learning. *In AAAI*, pages 9977–9978, 2019.
- [148] Shengchao LIU, Weili NIE, Chengpeng WANG, Jiarui LU, Zhuoran QIAO, Ling LIU, Jian TANG, Chaowei XIAO et Anima ANANDKUMAR : Multi-modal molecule structure-text model for text-based retrieval and editing. *arXiv preprint arXiv:2212.10789*, 2022.

- [149] Shengchao LIU, Weili NIE, Chengpeng WANG, Jiarui LU, Zhuoran QIAO, Ling LIU, Jian TANG, Chaowei XIAO et Anima ANANDKUMAR : Multi-modal molecule structure-text model for text-based retrieval and editing. *In Nature Machine Intelligence Under Review*, 2023.
- [150] Shengchao LIU, Meng QU, Zuobai ZHANG, Huiyu CAI et Jian TANG : Structured multi-task learning for molecular property prediction. *In AISTATS*, pages 8906–8920. PMLR, 2022.
- [151] Shengchao LIU, David VAZQUEZ, Jian TANG et Pierre-Andr e NO"EL : Flaky performances when pretraining on relational databases. *In AAAI*, 2023.
- [152] Shengchao LIU, Chengpeng WANG, Weili NIE, Hanchen WANG, Jiarui LU, Bolei ZHOU et Jian TANG : GraphCG: Unsupervised discovery of steerable factors in graphs. *In Submission to ICML*, 2023.
- [153] Shengchao LIU, Hanchen WANG, Weiyang LIU, Joan LASENBY, Hongyu GUO et Jian TANG : Pre-training molecular graph representation with 3d geometry. *arXiv preprint arXiv:2110.07728*, 2021.
- [154] Shengchao LIU, Hanchen WANG, Weiyang LIU, Joan LASENBY, Hongyu GUO et Jian TANG : Pre-training molecular graph representation with 3d geometry. *In ICLR*, 2022.
- [155] Shengchao LIU, Yutao ZHU, Jiarui LU, Zhao XU, Weili NIE, Anthony GITTER, Chaowei XIAO, Jian TANG, Hongyu GUO et Anima ANANDKUMAR : A text-guided protein design framework. *arXiv preprint arXiv:2302.04611*, 2023.
- [156] Shengchao LIU, Yutao ZHU, Jiarui LU, Zhao XU, Weili NIE, Anthony GITTER, Chaowei XIAO, Jian TANG, Hongyu GUO et Anima ANANDKUMAR : A text-guided protein design framework. *In Submission to ICML*, 2023.
- [157] Shikun LIU, Edward JOHNS et Andrew J DAVISON : End-to-end multi-task learning with attention. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1871–1880, 2019.
- [158] Xiao LIU, Fanjin ZHANG, Zhenyu HOU, Li MIAN, Zhaoyu WANG, Jing ZHANG et Jie TANG : Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering*, 2021.
- [159] Yi LIU, Limei WANG, Meng LIU, Xuan ZHANG, Bora OZTEKIN et Shuiwang JI : Spherical message passing for 3d graph networks. *arXiv preprint arXiv:2102.05013*, 2021.
- [160] Yixin LIU, Shirui PAN, Ming JIN, Chuan ZHOU, Feng XIA et Philip S YU : Graph self-supervised learning: A survey. *arXiv preprint arXiv:2103.00111*, 2021.
- [161] Yu-Shen LIU, Qi LI, Guo-Qin ZHENG, Karthik RAMANI et William BENJAMIN : Using diffusion distances for flexible molecular shape comparison. *BMC bioinformatics*, 11(1): 1–15, 2010.

- [162] Kyle LO, Lucy Lu WANG, Mark NEUMANN, Rodney KINNEY et Dan S WELD : S2orc: The semantic scholar open research corpus. *arXiv preprint arXiv:1911.02782*, 2019.
- [163] Ilya LOSHCHILOV et Frank HUTTER : Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.
- [164] Yongxi LU, Abhishek KUMAR, Shuangfei ZHAI, Yu CHENG, Tara JAVIDI et Rogerio FERIS : Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5334–5343, 2017.
- [165] Ines Filipa MARTINS, Ana L TEIXEIRA, Luis PINHEIRO et Andre O FALCAO : A bayesian approach to in silico blood-brain barrier penetration modeling. *Journal of chemical information and modeling*, 52(6):1686–1697, 2012.
- [166] Andreas MAYR, G"unter KLAMBAUER, Thomas UNTERTHINER, Marvin STEIJAERT, J"org K WEGNER, Hugo CEULEMANS, Djork-Arn'e CLEVERT et Sepp HOCHREITER : Large-scale comparison of machine learning methods for drug target prediction on chembl. *Chemical science*, 9(24):5441–5451, 2018.
- [167] RVN MELNIK, A UHLHERR, J HODGKIN et F DE HOOG : Distance geometry algorithms in molecular modelling of polymer and composite systems. *Computers & Mathematics with Applications*, 45(1-3):515–534, 2003.
- [168] David MENDEZ, Anna GAULTON, A Patrícia BENTO, Jon CHAMBERS, Marleen DE VEIJ, Eloy FÉLIX, María Paula MAGARIÑOS, Juan F MOSQUERA, Prudence MUTOWO, Michał NOWOTKA, María GORDILLO-MARAÑÓN, Fiona HUNTER, Laura JUNCO, Grace MUGUMBATE, Milagros RODRIGUEZ-LOPEZ, Francis ATKINSON, Nicolas BOSCH, Chris J RADOUX, Aldo SEGURA-CABRERA, Anne HERSEY et Andrew R LEACH : ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Research*, 47(D1):D930–D940, 11 2018.
- [169] MERCK : Merck molecular activity challenge. <https://www.kaggle.com/c/MerckActivity>, 2012.
- [170] Jesse G MEYER, Shengchao LIU, Ian J MILLER, Joshua J COON et Anthony GITTER : Learning drug functions from chemical structures with convolutional neural networks and random forests. *Journal of chemical information and modeling*, 59(10):4438–4449, 2019.
- [171] Ishan MISRA, Abhinav SHRIVASTAVA, Abhinav GUPTA et Martial HEBERT : Cross-stitch networks for multi-task learning. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3994–4003, 2016.
- [172] Andriy MNIH et Yee Whye TEH : A fast and simple algorithm for training neural probabilistic language models. *arXiv preprint arXiv:1206.6426*, 2012.
- [173] Gerry P MOSS : Basic terminology of stereochemistry (iupac recommendations 1996). *Pure and applied chemistry*, 68(12):2193–2222, 1996.

- [174] Maho NAKATA et Tomomi SHIMAZAKI : Pubchemqc project: a large-scale first-principles electronic structure database for data-driven chemistry. *Journal of chemical information and modeling*, 57(6):1300–1308, 2017.
- [175] Radford M NEAL et Geoffrey E HINTON : A view of the em algorithm that justifies incremental, sparse, and other variants. *In Learning in graphical models*, pages 355–368. Springer, 1998.
- [176] Thin NGUYEN, Hang LE et Svetha VENKATESH : Graphdta: prediction of drug–target binding affinity using graph convolutional networks. *BioRxiv*, page 684662, 2019.
- [177] Alex NICHOL, Prafulla DHARIWAL, Aditya RAMESH, Pranav SHYAM, Pamela MISHKIN, Bob MCGREW, Ilya SUTSKEVER et Mark CHEN : Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.
- [178] Didrik NIELSEN, Priyank JAINI, Emiel HOOGEBOOM, Ole WINTHER et Max WELLING : Survae flows: Surjections to bridge the gap between vaes and flows. *Advances in Neural Information Processing Systems*, 33, 2020.
- [179] Erik NIJKAMP, Ruiqi GAO, Pavel SOUNTSOV, Srinivas VASUDEVAN, Bo PANG, Song-Chun ZHU et Ying Nian WU : Learning energy-based model with flow-based backbone by neural transport mcmc. *arXiv preprint arXiv:2006.06897*, 2020.
- [180] Sebastian NOWOZIN, Botond CSEKE et Ryota TOMIOKA : f-gan: Training generative neural samplers using variational divergence minimization. *In Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 271–279, 2016.
- [181] Aaron van den OORD, Yazhe LI et Oriol VINYALS : Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [182] Hakime "OZT"URK, Arzucan "OZG"UR et Elif OZKIRIMLI : Deepdta: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.
- [183] Tapio PAHIKKALA, Antti AIROLA, Sami PIETIL"A, Sushil SHAKYAWAR, Agnieszka SZWAJDA, Jing TANG et Tero AITTOKALLIO : Toward more realistic drug–target interaction predictions. *Briefings in bioinformatics*, 16(2):325–337, 2015.
- [184] Or PATASHNIK, Zongze WU, Eli SHECHTMAN, Daniel COHEN-OR et Dani LISCHINSKI : Styleclip: Text-driven manipulation of stylegan imagery. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2085–2094, 2021.
- [185] Atanas PATRONOV, Kostas PAPAPOPOULOS et Ola ENGVIST : Has artificial intelligence impacted drug discovery? *In Artificial Intelligence in Drug Design*, pages 153–176. Springer, 2022.
- [186] Lagnajit PATTANAIK, Octavian-Eugen GANEA, Ian COLEY, Klavs F JENSEN, William H GREEN et Connor W COLEY : Message passing networks for molecules with tetrahedral chirality. *arXiv preprint arXiv:2012.00094*, 2020.

- [187] Francesca PISTILLI, Giulia FRACASTORO, Diego VALSESIA et Enrico MAGLI : Learning graph-convolutional representations for point cloud denoising. *In European conference on computer vision*, pages 103–118. Springer, 2020.
- [188] Ben POOLE, Sherjil OZAIR, Aaron VAN DEN OORD, Alex ALEMI et George TUCKER : On variational bounds of mutual information. *In International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.
- [189] Charles Ruizhongtai QI, Li YI, Hao SU et Leonidas J GUIBAS : Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.
- [190] Zhuoran QIAO, Anders S CHRISTENSEN, Frederick R MANBY, Matthew WELBORN, Anima ANANDKUMAR et Thomas F MILLER III : Unite: Unitary n-body tensor equivariant network with applications to quantum chemistry. *arXiv preprint arXiv:2105.14655*, 2021.
- [191] Alec RADFORD, Jong Wook KIM, Chris HALLACY, Aditya RAMESH, Gabriel GOH, Sandhini AGARWAL, Girish SASTRY, Amanda ASKELL, Pamela MISHKIN, Jack CLARK et al. : Learning transferable visual models from natural language supervision. *In International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.
- [192] Raghunathan RAMAKRISHNAN, Pavlo O DRAL, Matthias RUPP et O Anatole VON LILIENFELD : Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.
- [193] Aditya RAMESH, Prafulla DHARIWAL, Alex NICHOL, Casey CHU et Mark CHEN : Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.
- [194] Ladislav RAMP’ASEK, Mikhail GALKIN, Vijay Prakash DWIVEDI, Anh Tuan LUU, Guy WOLF et Dominique BEAINI : Recipe for a general, powerful, scalable graph transformer. *arXiv preprint arXiv:2205.12454*, 2022.
- [195] Bharath RAMSUNDAR, Steven KEARNES, Patrick RILEY, Dale WEBSTER, David KONERDING et Vijay PANDE : Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072*, 2015.
- [196] Bharath RAMSUNDAR, Bowen LIU, Zhenqin WU, Andreas VERRAS, Matthew TUDOR, Robert P SHERIDAN et Vijay PANDE : Is multitask deep learning practical for pharma? *Journal of chemical information and modeling*, 57(8):2068–2076, 2017.
- [197] R Leila REYNALD, Stefaan SANSEN, C David STOUT et Eric F JOHNSON : Structural characterization of human cytochrome p450 2c19: Active site differences between p450s 2c8, 2c9, and 2c19. *Journal of Biological Chemistry*, 287(53):44581–44591, 2012.
- [198] Sebastian G. ROHRER et Knut BAUMANN : Maximum unbiased validation (muv) data sets for virtual screening based on pubchem bioactivity data. *Journal of Chemical Information and Modeling*, 49(2):169–184, 2009. PMID: 19161251.

- [199] Sebastian G ROHRER et Knut BAUMANN : Maximum unbiased validation (muv) data sets for virtual screening based on pubchem bioactivity data. *Journal of chemical information and modeling*, 49(2):169–184, 2009.
- [200] Yu RONG, Yatao BIAN, Tingyang XU, Weiyang XIE, Ying WEI, Wenbing HUANG et Junzhou HUANG : Self-supervised graph transformer on large-scale molecular data. *In Advances in Neural Information Processing Systems, NeurIPS*, 2020.
- [201] Chitwan SAHARIA, William CHAN, Saurabh SAXENA, Lala LI, Jay WHANG, Emily DENTON, Seyed Kamyar Seyed GHASEMIPOUR, Burcu Karagol AYAN, S Sara MAHDAVI, Rapha Gontijo LOPES *et al.* : Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.
- [202] Victor Garcia SATORRAS, Emiel HOOGEBOOM et Max WELLING : E (n) equivariant graph neural networks. *arXiv preprint arXiv:2102.09844*, 2021.
- [203] Nikunj SAUNSHI, Orestis PLEVRAKIS, Sanjeev ARORA, Mikhail KHODAK et Hrishikesh KHANDEPARKAR : A theoretical analysis of contrastive unsupervised representation learning. *In ICML*, pages 5628–5637. PMLR, 2019.
- [204] H Bernhard SCHLEGEL : Exploring potential energy surfaces for chemical reactions: an overview of some practical methods. *Journal of computational chemistry*, 24(12):1514–1527, 2003.
- [205] Rodney Caughren SCHNUR et Lee Daniel ARNOLD : Alkynyl and azido-substituted 4-anilinoquinazolines, mai 1998. US Patent 5,747,498.
- [206] Kristof T SCH"UTT, Pieter-Jan KINDERMANS, Huziel E SAUCEDA, Stefan CHMIELA, Alexandre TKATCHENKO et Klaus-Robert M"ULLER : Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *arXiv preprint arXiv:1706.08566*, 2017.
- [207] Kristof T SCH"UTT, Huziel E SAUCEDA, P-J KINDERMANS, Alexandre TKATCHENKO et K-R M"ULLER : Schnet—a deep learning architecture for molecules and materials. *The Journal of Chemical Physics*, 148(24):241722, 2018.
- [208] Kristof T SCH"UTT, Oliver T UNKE et Michael GASTEGGER : Equivariant message passing for the prediction of tensorial properties and molecular spectra. *arXiv preprint arXiv:2102.03150*, 2021.
- [209] Chence SHI, Shitong LUO, Minkai XU et Jian TANG : Learning gradient fields for molecular conformation generation. *In International Conference on Machine Learning*, pages 9558–9568. PMLR, 2021.
- [210] Chence SHI, Minkai XU, Hongyu GUO, Ming ZHANG et Jian TANG : A graph to graphs framework for retrosynthesis prediction. *In International Conference on Machine Learning*, pages 8818–8827. PMLR, 2020.
- [211] Chence SHI, Minkai XU, Zhaocheng ZHU, Weinan ZHANG, Ming ZHANG et Jian TANG : Graphaf: a flow-based autoregressive model for molecular graph generation. *arXiv*

- preprint arXiv:2001.09382*, 2020.
- [212] Weijing SHI et Raj RAJKUMAR : Point-gnn: Graph neural network for 3d object detection in a point cloud. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1711–1719, 2020.
- [213] Yunsheng SHI, Zhengjie HUANG, Wenjin WANG, Hui ZHONG, Shikun FENG et Yu SUN : Masked label prediction: Unified message passing model for semi-supervised classification. *arXiv preprint arXiv:2009.03509*, 2020.
- [214] Muhammed SHUAIBI, Adeesh KOLLURU, Abhishek DAS, Aditya GROVER, Anuroop SRIRAM, Zachary ULISSI et C Lawrence ZITNICK : Rotation invariant graph neural networks using spin convolutions. *arXiv preprint arXiv:2106.09575*, 2021.
- [215] Anders SØGAARD et Yoav GOLDBERG : Deep multi-task learning with low level tasks supervised at lower layers. *In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 231–235, Berlin, Germany, août 2016. Association for Computational Linguistics.
- [216] Jascha SOHL-DICKSTEIN, Eric WEISS, Niru MAHESWARANATHAN et Surya GANGULI : Deep unsupervised learning using nonequilibrium thermodynamics. *In International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.
- [217] Gowthami SOMEPELLI, Micah GOLDBLUM, Avi SCHWARZSCHILD, C Bayan BRUSS et Tom GOLDSTEIN : Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. *arXiv preprint arXiv:2106.01342*, 2021.
- [218] Yang SONG et Stefano ERMON : Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.
- [219] Yang SONG et Stefano ERMON : Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.
- [220] Yang SONG et Diederik P KINGMA : How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.
- [221] Yang SONG, Jascha SOHL-DICKSTEIN, Diederik P KINGMA, Abhishek KUMAR, Stefano ERMON et Ben POOLE : Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [222] Antonia STANK, Daria B KOKH, Jonathan C FULLER et Rebecca C WADE : Protein binding pocket dynamics. *Accounts of chemical research*, 49(5):809–815, 2016.
- [223] Hannes STARK, Dominique BEAINI, Gabriele CORSO, Prudencio TOSSOU, Christian DALLAGO, Stephan GUNNEMANN et Pietro LIÒ : 3d infomax improves gnns for molecular property prediction. *In International Conference on Machine Learning*, pages 20479–20502. PMLR, 2022.
- [224] Teague STERLING et John J IRWIN : Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

- [225] Hachiro SUGIMOTO, Youichi IIMURA, Yoshiharu YAMANISHI et Kiyomi YAMATSU : Synthesis and structure-activity relationships of acetylcholinesterase inhibitors: 1-benzyl-4-[(5,6-dimethoxy-1-oxoindan-2-yl)methyl]piperidine hydrochloride and related compounds. *Journal of Medicinal Chemistry*, 38(24):4821–4829, 1995. PMID: 7490731.
- [226] Thomas SULLIVAN : A tough road: cost to develop one new drug is \$2.6 billion; approval rate for drugs entering clinical development is less than 12%. *Policy & Medicine*, 2019.
- [227] Fan-Yun SUN, Jordan HOFFMANN, Vikas VERMA et Jian TANG : Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. *In International Conference on Learning Representations, ICLR*, 2020.
- [228] Ruoxi SUN : Does gnn pretraining help molecular representation? *NeurIPS*, 2022.
- [229] Ruoxi SUN, Hanjun DAI, Li LI, Steven KEARNES et Bo DAI : Energy-based view of retrosynthesis. *arXiv preprint arXiv:2007.13437*, 2020.
- [230] Ruoxi SUN, Hanjun DAI et Adams Wei YU : Rethinking of graph pretraining on molecular representation. 2022.
- [231] Damian SZKLARCZYK, Annika L GABLE, David LYON, Alexander JUNGE, Stefan WYDER, Jaime HUERTA-CEPAS, Milan SIMONOVIC, Nadezhda T DONCHEVA, John H MORRIS, Peer BORK *et al.* : String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic acids research*, 47(D1):D607–D613, 2019.
- [232] John J TALLEY, Thomas D PENNING, Paul W COLLINS, Donald J ROGIER JR, James W MALECHA, Julie M MIYASHIRO, Stephen R BERTENSHAW, Ish K KHANNA, Matthew J GRANETO, Roland S ROGERS *et al.* : Substituted pyrazolyl benzenesulfonamides for the treatment of inflammation, juin 1998. US Patent 5,760,068.
- [233] Jing TANG, Agnieszka SZWAJDA, Sushil SHAKYAWAR, Tao XU, Petteri HINTSANEN, Krister WENNERBERG et Tero AITTOKALLIO : Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. *Journal of Chemical Information and Modeling*, 54(3):735–743, 2014.
- [234] Nathaniel THOMAS, Tess SMIDT, Steven KEARNES, Lusann YANG, Li LI, Kai KOHLHOFF et Patrick RILEY : Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018.
- [235] Yuandong TIAN, Xinlei CHEN et Surya GANGULI : Understanding self-supervised learning dynamics without contrastive pairs. *In ICML*, pages 10268–10278. PMLR, 2021.
- [236] Raphael JL TOWNSHEND, Martin V"OGELE, Patricia SURIANA, Alexander DERRY, Alexander POWERS, Yianni LALOUDAKIS, Sidhika BALACHANDAR, Brandon ANDERSON, Stephan EISMANN, Risi KONDOR *et al.* : Atom3d: Tasks on molecules in three

- dimensions. *arXiv preprint arXiv:2012.04035*, 2020.
- [237] TOX21 DATA CHALLENGE : Tox21 data challenge 2014. <https://tripod.nih.gov/tox21/challenge/>, 2014.
- [238] Oleg TROTT et Arthur J OLSON : Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461, 2010.
- [239] Michael TSCHANNEN, Josip DJOLONGA, Paul K RUBENSTEIN, Sylvain GELLY et Mario LUCIC : On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.
- [240] Thomas UNTERTHINER, Andreas MAYR, G"unter KLAMBAUER, Marvin STEIJAERT, J"org K WEGNER, Hugo CEULEMANS et Sepp HOCHREITER : Deep learning as an opportunity in virtual screening. *Advances in neural information processing systems*, 27, 2014.
- [241] Mikaela Angelina UY, Quang-Hieu PHAM, Binh-Son HUA, Thanh NGUYEN et Sai-Kit YEUNG : Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. *In Proceedings of the IEEE/CVF international conference on computer vision*, pages 1588–1597, 2019.
- [242] Aaron Van den OORD, Yazhe LI et Oriol VINYALS : Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807, 2018.
- [243] Vladimir VAPNIK : *The nature of statistical learning theory*. Springer science & business media, 2013.
- [244] Vladimir VAPNIK, Rauf IZMAILOV *et al.* : Learning using privileged information: similarity control and knowledge transfer. *J. Mach. Learn. Res.*, 16(1):2023–2049, 2015.
- [245] Vladimir VAPNIK et Akshay VASHIST : A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557, 2009.
- [246] Ashish VASWANI, Noam SHAZEER, Niki PARMAR, Jakob USZKOREIT, Llion JONES, Aidan N GOMEZ, Łukasz KAISER et Illia POLOSUKHIN : Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [247] Petar VELICKOVI'Ć, William FEDUS, William L HAMILTON, Pietro LIÒ, Yoshua BENGIO et R Devon HJELM : Deep graph infomax. *arXiv preprint arXiv:1809.10341*, 2018.
- [248] Pascal VINCENT : A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.
- [249] Pascal VINCENT, Hugo LAROCHELLE, Yoshua BENGIO et Pierre-Antoine MANZAGOL : Extracting and composing robust features with denoising autoencoders. *In Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

- [250] Hanchen WANG, Jean KADDOUR, Shengchao LIU, Jian TANG, Matt KUSNER, Joan LASENBY et Qi LIU : Evaluating self-supervised learning for molecular graph embeddings, 2022.
- [251] Hanchen WANG, Qi LIU, Xiangyu YUE, Joan LASENBY et Matthew J. KUSNER : Unsupervised point cloud pre-training via view-point occlusion, completion. *In ICCV*, 2021.
- [252] Renxiao WANG, Xueliang FANG, Yipin LU, Chao-Yie YANG et Shaomeng WANG : The pdbbind database: methodologies and updates. *Journal of medicinal chemistry*, 48(12):4111–4119, 2005.
- [253] Tongzhou WANG et Phillip ISOLA : Understanding contrastive representation learning through alignment and uniformity on the hypersphere. *In International Conference on Machine Learning, ICML*, 2020.
- [254] Yingheng WANG, Yaosen MIN, Xin CHEN et Ji WU : Multi-view graph contrastive representation learning for drug-drug interaction prediction. *In Proceedings of the Web Conference 2021*, pages 2921–2933, 2021.
- [255] Yuyang WANG, Jianren WANG, Zhonglin CAO et Amir BARATI FARIMANI : Molecular contrastive learning of representations via graph neural networks. *Nature Machine Intelligence*, 4(3):279–287, 2022.
- [256] Yuyang WANG, Jianren WANG, Zhonglin CAO et Amir Barati FARIMANI : Molclr: Molecular contrastive learning of representations via graph neural networks. *arXiv preprint arXiv:2102.10056*, 2021.
- [257] Zichao WANG, Weili NIE, Zhuoran QIAO, Chaowei XIAO, Richard BARANIUK et Anima ANANDKUMAR : Retrieval-based controllable molecule generation. *arXiv preprint arXiv:2208.11126*, 2022.
- [258] Zirui WANG, Yulia TSVETKOV, Orhan FIRAT et Yuan CAO : Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. *In International Conference on Learning Representations*, 2021.
- [259] David WEININGER : Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.
- [260] Ming WEN, Zhimin ZHANG, Shaoyu NIU, Haozhi SHA, Ruihan YANG, Yonghuan YUN et Hongmei LU : Deep-learning-based drug–target interaction prediction. *Journal of proteome research*, 16(4):1401–1409, 2017.
- [261] David S WISHART, Yannick D FEUNANG, An C GUO, Elvis J LO, Ana MARCU, Jason R GRANT, Tanvir SAJED, Daniel JOHNSON, Carin LI, Zinat SAYEEDA *et al.* : Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, 46(D1):D1074–D1082, 2018.

- [262] Lirong WU, Haitao LIN, Zhangyang GAO, Cheng TAN, Stan LI *et al.* : Self-supervised on graphs: Contrastive, generative, or predictive. *arXiv preprint arXiv:2105.07342*, 2021.
- [263] Zhenqin WU, Bharath RAMSUNDAR, Evan N FEINBERG, Joseph GOMES, Caleb GENIESSE, Aneesh S PAPPU, Karl LESWING et Vijay PANDE : Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.
- [264] Jianwen XIE, Yang LU, Song-Chun ZHU et Yingnian WU : A theory of generative convnet. *In International Conference on Machine Learning*, pages 2635–2644. PMLR, 2016.
- [265] Yaochen XIE, Zhao XU, Jingtun ZHANG, Zhengyang WANG et Shuiwang JI : Self-supervised learning of graph neural networks: A unified review. *arXiv preprint arXiv:2102.10757*, 2021.
- [266] Keyulu XU, Weihua HU, Jure LESKOVEC et Stefanie JEGELKA : How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.
- [267] Minghao XU, Hang WANG, Bingbing NI, Hongyu GUO et Jian TANG : Self-supervised graph-level representation learning with local and global structure. *In International Conference on Machine Learning, ICML*, 2021.
- [268] Zhao XU, Youzhi LUO, Xuan ZHANG, Xinyi XU, Yaochen XIE, Meng LIU, Kaleb DICKERSON, Cheng DENG, Maho NAKATA et Shuiwang JI : Molecule3d: A benchmark for predicting 3d geometries from molecular graphs. *arXiv preprint arXiv:2110.01717*, 2021.
- [269] Zhao XU, Youzhi LUO, Xuan ZHANG, Xinyi XU, Yaochen XIE, Meng LIU, Kaleb Andrew DICKERSON, Cheng DENG, Maho NAKATA et Shuiwang JI : Molecule3d: A benchmark for predicting 3d geometries from molecular graphs, 2021.
- [270] Kevin YANG, Kyle SWANSON, Wengong JIN, Connor COLEY, Philipp EIDEN, Hua GAO, Angel GUZMAN-PEREZ, Timothy HOPPER, Brian KELLEY, Miriam MATHEA *et al.* : Analyzing learned molecular representations for property prediction. *Journal of chemical information and modeling*, 59(8):3370–3388, 2019.
- [271] Huaxiu YAO, Xinyu YANG, Xinyi PAN, Shengchao LIU, Pang Wei KOH et Chelsea FINN : Leveraging domain relations for domain generalization. *arXiv preprint arXiv:2302.02609*, 2023.
- [272] Chengxuan YING, Tianle CAI, Shengjie LUO, Shuxin ZHENG, Guolin KE, Di HE, Yanming SHEN et Tie-Yan LIU : Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems*, 34, 2021.
- [273] Yuning YOU, Tianlong CHEN, Yang SHEN et Zhangyang WANG : Graph contrastive learning automated. *In International Conference on Machine Learning, ICML*, 2021.

- [274] Yuning YOU, Tianlong CHEN, Yongduo SUI, Ting CHEN, Zhangyang WANG et Yang SHEN : Graph contrastive learning with augmentations. *In Advances in Neural Information Processing Systems, NeurIPS*, 2020.
- [275] Yuning YOU, Tianlong CHEN, Yongduo SUI, Ting CHEN, Zhangyang WANG et Yang SHEN : Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823, 2020.
- [276] Tianhe YU, Saurabh KUMAR, Abhishek GUPTA, Sergey LEVINE, Karol HAUSMAN et Chelsea FINN : Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020.
- [277] Daniel ZAHAREVITZ : Aids antiviral screen data, 2015.
- [278] Sheheryar ZAIDI, Michael SCHAARSCHMIDT, James MARTENS, Hyunjik KIM, Yee Whye TEH, Alvaro SANCHEZ-GONZALEZ, Peter BATTAGLIA, Razvan PASCANU et Jonathan GODWIN : Pre-training via denoising for molecular property prediction. *arXiv preprint arXiv:2206.00133*, 2022.
- [279] Chengxi ZANG et Fei WANG : Moflow: an invertible flow model for generating molecular graphs. *In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 617–626, 2020.
- [280] Zheni ZENG, Yuan YAO, Zhiyuan LIU et Maosong SUN : A deep-learning system bridging molecule structure and biomedical text with comprehension comparable to human professionals. *Nature communications*, 13(1):1–11, 2022.
- [281] Ningyu ZHANG, Zhen BI, Xiaozhuan LIANG, Siyuan CHENG, Haosen HONG, Shumin DENG, Jiazhang LIAN, Qiang ZHANG et Huajun CHEN : Ontoprotein: Protein pre-training with gene ontology embedding. *arXiv preprint arXiv:2201.11147*, 2022.
- [282] Yu ZHANG, Pengyuan ZHANG et Yonghong YAN : Attention-based lstm with multi-task learning for distant speech recognition. *In Interspeech*, pages 3857–3861, 2017.
- [283] Alex ZHAVORONKOV, Yan A IVANENKOV, Alex ALIPER, Mark S VESELOV, Vladimir A ALADINSKIY, Anastasiya V ALADINSKAYA, Victor A TERENCEV, Daniil A POLYKOVSKIY, Maksim D KUZNETSOV, Arip ASADULAEV *et al.* : Deep learning enables rapid identification of potent ddr1 kinase inhibitors. *Nature biotechnology*, 37(9):1038–1040, 2019.
- [284] Jinhua ZHU, Yingce XIA, Chang LIU, Lijun WU, Shufang XIE, Tong WANG, Yusong WANG, Wengang ZHOU, Tao QIN, Houqiang LI *et al.* : Direct molecular conformation generation. *arXiv preprint arXiv:2202.01356*, 2022.
- [285] Jinhua ZHU, Yingce XIA, Tao QIN, Wengang ZHOU, Houqiang LI et Tie-Yan LIU : Dual-view molecule pre-training. *arXiv preprint arXiv:2106.10234*, 2021.

Appendix A

Appendix for GraphMVP: Pre-training Molecular Graph Representation with 3D Geometry

A.1. Self-Supervised Learning on Molecular Graph

Self-supervised learning (SSL) methods have attracted massive attention recently, trending from vision [26, 28, 30, 87, 251], language [24, 45, 181] to graph [99, 140, 227, 247, 273, 274]. In general, there are two categories of SSL: contrastive and generative, where they differ on the design of the supervised signals. Contrastive SSL realizes the supervised signals at the **inter-data** level, learning the representation by contrasting with other data points; while generative SSL focuses on reconstructing the original data at the **intra-data** level. Both venues have been widely explored [158, 160, 262, 265].

A.1.1. Contrastive graph SSL

Contrastive graph SSL first applies transformations to construct different *views* for each graph. Each view incorporates different granularities of information, like node-, subgraph-, and graph-level. It then solves two sub-tasks simultaneously: (1) aligning the representations of views from the same data; (2) contrasting the representations of views from different data, leading to a uniformly distributed latent space [253]. The key difference among existing methods is thus the design of view constructions. InfoGraph [227, 247] contrasted the node (local) and graph (global) views. ContextPred [99] and G-Contextual [200] contrasted between node and context views. GraphCL and JOAO [273, 274] made comprehensive comparisons among four graph-level transformations and further learned to select the most effective combinations.

A.1.2. Generative graph SSL

Generative graph SSL aims at reconstructing important structures for each graph. By so doing, it consequently learns a representation capable of encoding key ingredients of the data. EdgePred [82] and AttrMask [99] predicted the adjacency matrix and masked tokens (nodes and edges) respectively. GPT-GNN [101] reconstructed the whole graph in an auto-regressive approach.

A.1.3. Predictive graph SSL

There are certain SSL methods specific to the molecular graph. For example, one central task in drug discovery is to find the important substructure or motif in molecules that can activate the target interactions. G-Motif [200] adopts domain knowledge to heuristically extract motifs for each molecule, and the SSL task is to make prediction on the existence of each motif. Different from contrastive and generative SSL, recent literature [262] takes this as predictive graph SSL, where the supervised signals are self-generated labels.

SSL for Molecular Graphs. Recall that all previous methods in Table 19 **merely** focus on the 2D topology. However, for science-centric tasks such as molecular property prediction, 3D geometry should be incorporated as it provides complementary and comprehensive information [159, 206]. To mitigate this gap, we propose GraphMVP to leverage the 3D geometry with unsupervised graph pre-training.

A.2. Molecular Graph Representation

There are two main methods for molecular graph representation learning. The first one is the molecular fingerprints. It is a hashed bit vector to describe the molecular graph. There

Table 19. Comparison between GraphMVP and existing graph SSL methods.

SSL Pre-training	Graph View		SSL Category		
	2D Topology	3D Geometry	Generative	Contrastive	Predictive
EdgePred [82]	✓	-	✓	-	-
AttrMask [99]	✓	-	✓	-	-
GPT-GNN [101]	✓	-	✓	-	-
InfoGraph [227, 247]	✓	-	-	✓	-
ContexPred [99]	✓	-	-	✓	-
GraphLoG [267]	✓	-	-	✓	-
G-Contextual [200]	✓	-	-	✓	-
GraphCL [274]	✓	-	-	✓	-
JOAO [273]	✓	-	-	✓	-
G-Motif [200]	✓	-	-	-	✓
GraphMVP(Ours)	✓	✓	✓	✓	-

has been re-discoveries on fingerprints-based methods [2, 111, 138, 147, 170, 195], while it has one main drawback: Random forest and XGBoost are very strong learning models on fingerprints, but they fail to take benefits of the pre-training strategy.

Graph neural network (GNN) has become another mainstream modeling methods for molecular graph representation. Existing methods can be generally split into two venues: 2D GNN and 3D GNN, depending on what levels of information is considered. 2D GNN focuses on the topological structures of the graph, like the adjacency among nodes, while 3D GNN is able to model the “energy” of molecules by taking account the spatial positions of atoms.

First, we want to highlight that GraphMVP is model-agnostic, *i.e.*, it can be applied to any 2D and 3D GNN representation function, yet the specific 2D and 3D representations are not the main focus of this work. Second, we acknowledge there are a lot of advanced 3D [61, 115, 159, 202] and 2D [36, 43, 69, 140, 266, 270] representation methods. However, considering the *graph SSL literature* and *graph representation literature* (illustrated below), we adopt GIN [266] and SchNet [206] in current GraphMVP.

A.2.1. 2D Molecular Graph Neural Network

The 2D representation is taking each molecule as a 2D graph, with atoms as nodes and bonds as edges, *i.e.*, $g_{2D} = (X, E)$. $X \in \mathbb{R}^{n \times d_n}$ is the atom attribute matrix, where n is the number of atoms (nodes) and d_n is the atom attribute dimension. $E \in \mathbb{R}^{m \times d_e}$ is the bond attribute matrix, where m is the number of bonds (edges) and d_m is the bond attribute dimension. Notice that here E also includes the connectivity. Then we will apply a transformation function T_{2D} on the topological graph. Given a 2D graph g_{2D} , its 2D molecular representation is:

$$h_{2D} = \text{GNN-2D}(T_{2D}(g_{2D})) = \text{GNN-2D}(T_{2D}(X, E)). \quad (\text{A.2.1})$$

The core operation of 2D GNN is the message passing function [69], which updates the node representation based on adjacency information. We have variants depending on the design of message and aggregation functions, and we pick GIN [266] in this work.

GIN. There has been a long research line on 2D graph representation learning [36, 43, 69, 140, 266, 270]. Among these, graph isomorphism network (GIN) model [266] has been widely used as the backbone model in recent graph self-supervised learning work [99, 273, 274]. Thus, we as well adopt GIN as the base model for 2D representation.

Recall each molecule is represented as a molecular graph, *i.e.*, $g_{2D} = (X, E)$, where X and E are feature matrices for atoms and bonds respectively. Then the message passing

function is defined as:

$$z_i^{(k+1)} = \text{MLP}_{\text{atom}}^{(k+1)}\left(z_i^{(k)} + \sum_{j \in \mathcal{N}(i)} \left(z_j^{(k)} + \text{MLP}_{\text{bond}}^{(k+1)}(E_{ij})\right)\right), \quad (\text{A.2.2})$$

where $z_0 = X$ and $\text{MLP}_{\text{atom}}^{(k+1)}$ and $\text{MLP}_{\text{bond}}^{(k+1)}$ are the $(l + 1)$ -th MLP layers on the atom- and bond-level respectively. Repeating this for K times, and we can encode K -hop neighborhood information for each center atom in the molecular data, and we take the last layer for each node/atom representation. The graph-level molecular representation is the mean of the node representation:

$$z(\mathbf{x}) = \frac{1}{N} \sum_i z_i^{(K)} \quad (\text{A.2.3})$$

A.2.2. 3D Molecular Graph Neural Network

Recently, the 3D geometric representation learning has brought breakthrough progress in molecule modeling [61, 115, 159, 202, 206]. 3D molecular graph additionally includes spatial locations of the atoms, which needless to be static since, in real scenarios, atoms are in continual motion on a *potential energy surface* [7]. The 3D structures at the local minima on this surface are named *molecular conformation* or *conformer*. As the molecular properties are a function of the conformer ensembles [85], this reveals another limitation of existing mainstream methods: to predict properties from a single 2D or 3D graph cannot account for this fact [7], while our proposed method can alleviate this issue to a certain extent.

For specific 3D molecular graph, it additionally includes spatial positions of the atoms. We represent each conformer as $g_{3\text{D}} = (X, R)$, where $R \in \mathbb{R}^{n \times 3}$ is the 3D-coordinate matrix, and the corresponding representation is:

$$h_{3\text{D}} = \text{GNN-3D}(T_{3\text{D}}(g_{3\text{D}})) = \text{GNN-3D}(T_{3\text{D}}(X, R)), \quad (\text{A.2.4})$$

where R is the 3D-coordinate matrix and $T_{3\text{D}}$ is the 3D transformation. Note that further information such as plane and torsion angles can be solved from the positions.

SchNet. SchNet [206] is composed of the following key steps:

$$\begin{aligned} z_i^{(0)} &= \text{embedding}(x_i) \\ z_i^{(t+1)} &= \text{MLP}\left(\sum_{j=1}^n f(x_j^{(t-1)}, r_i, r_j)\right) \\ h_i &= \text{MLP}(z_i^{(K)}), \end{aligned} \quad (\text{A.2.5})$$

where K is the number of hidden layers, and

$$f(x_j, r_i, r_j) = x_j \cdot e_k(r_i - r_j) = x_j \cdot \exp(-\gamma \|\|r_i - r_j\|_2 - \mu\|_2^2) \quad (\text{A.2.6})$$

is the continuous-filter convolution layer, enabling the modeling of continuous positions of atoms.

We adopt SchNet for the following reasons. (1) SchNet is a very strong geometric representation method after *fair* benchmarking. (2) SchNet can be trained more efficiently, comparing to the other recent 3D models. To support these two points, we make a comparison among the most recent 3D geometric models [61, 159, 202] on QM9 dataset. QM9 [263] is a molecule dataset approximating 12 thermodynamic properties calculated by density functional theory (DFT) algorithm. Notice: UNiTE [190] is the state-of-the-art 3D GNN, but it requires a commercial software for feature extraction, thus we exclude it for now.

Table 20. Reproduced MAE on QM9. 100k for training, 17,748 for val, 13,083 for test. The last column is the approximated running time.

	alpha	gap	homo	lumo	mu	cv	g298	h298	r2	u298	u0	zpve	time
SchNet [206]	0.077	50	32	26	0.030	0.032	15	14	0.122	14	14	1.751	3h
SE(3)-Trans [61]	0.143	59	36	36	0.052	0.068	68	72	1.969	68	74	5.517	50h
EGNN [202]	0.075	49	29	26	0.030	0.032	11	10	0.076	10	10	1.562	24h
SphereNet [159]	0.054	41	22	19	0.028	0.027	10	8	0.295	8	8	1.401	50h

Table 20 shows that, under a fair comparison (w.r.t. data splitting, seed, cuda version, etc), SchNet can reach pretty comparable performance, yet the efficiency of SchNet is much better. Combining these two points, we adopt SchNet in current version of GraphMVP.

A.2.3. Summary

To sum up, in GraphMVP, the most important message we want to deliver is how to design a well-motivated SSL algorithm to extract useful 3D geometry information to augment the 2D representation for downstream fine-tuning. GraphMVP is model-agnostic, and we may as well leave the more advanced 3D [61, 115, 159, 202] and 2D [36, 140, 270] GNN for future exploration.

In addition, molecular property prediction tasks have rich alternative representation methods, including SMILES [90, 259], and biological knowledge graph [150, 254]. There have been another SSL research line on them [59, 139, 285], yet they are beyond the scope of discussion in this paper.

A.3. Maximize Mutual Information

In what follows, we will use X and Y to denote the data space for 2D graph and 3D graph respectively. Then the latent representations are denoted as h_x and h_y .

A.3.1. Formulation

The standard formulation for mutual information (MI) is

$$I(X; Y) = \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} \left[\log \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} \right]. \tag{A.3.1}$$

Another well-explained MI inspired from wikipedia is given in Figure 14.

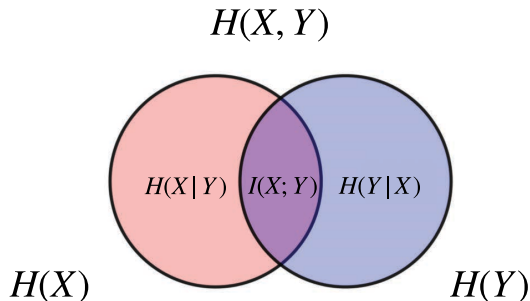


Figure 14. Venn diagram of mutual information. Inspired by wikipedia.

Mutual information (MI) between random variables measures the corresponding non-linear dependence. As can be seen in the first equation in Equation (A.3.1), the larger the divergence between the joint $p(\mathbf{x}, \mathbf{y})$ and the product of the marginals $p(\mathbf{x})p(\mathbf{y})$, the stronger the dependence between X and Y .

Thus, following this logic, maximizing MI between 2D and 3D views can force the 3D/2D representation to capture higher-level factors, *e.g.*, the occurrence of important substructure that is semantically vital for downstream tasks. Or equivalently, maximizing MI can decrease the uncertainty in 2D representation given 3D geometric information.

A.3.2. A Lower Bound to MI

To solve MI, we first extract a lower bound:

$$\begin{aligned} I(X; Y) &= \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} \left[\log \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} \right] \\ &\geq \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} \left[\log \frac{p(\mathbf{x}, \mathbf{y})}{\sqrt{p(\mathbf{x})p(\mathbf{y})}} \right] \\ &= \frac{1}{2} \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} \left[\log \frac{(p(\mathbf{x}, \mathbf{y}))^2}{p(\mathbf{x})p(\mathbf{y})} \right] \tag{A.3.2} \\ &= \frac{1}{2} \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} \left[\log p(\mathbf{x}|\mathbf{y}) \right] + \frac{1}{2} \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} \left[\log p(\mathbf{y}|\mathbf{x}) \right] \\ &= -\frac{1}{2} [H(Y|X) + H(X|Y)]. \end{aligned}$$

Thus, we transform the MI maximization problem into minimizing the following objective:

$$\mathcal{L}_{\text{MI}} = \frac{1}{2} [H(Y|X) + H(X|Y)]. \tag{A.3.3}$$

In the following sections, we will describe two self-supervised learning methods for solving MI. Notice that the methods are very general, and can be applied to various applications. Here we apply it mainly for making 3D geometry useful for 2D representation learning on molecules.

A.4. Contrastive Self-Supervised Learning

The essence of contrastive self-supervised learning is to align positive view pairs and contrast negative view pairs, such that the obtained representation space is well distributed [253]. We display the pipeline in Figure 15. Along the research line in graph SSL [158, 160, 262, 265], InfoNCE and EBM-NCE are the two most-widely used, as discussed below.

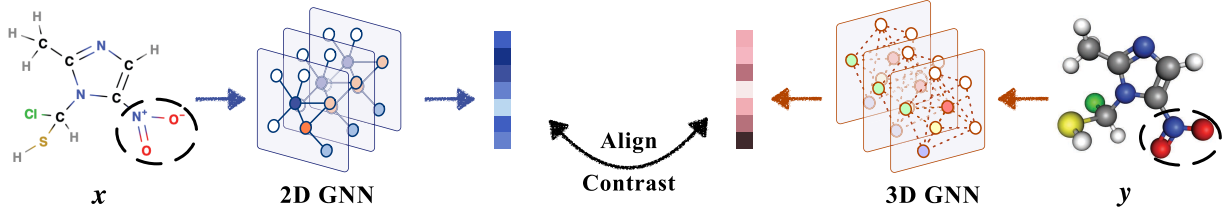


Figure 15. Contrastive SSL in GraphMVP. The black dashed circles represent subgraph masking.

A.4.1. InfoNCE

InfoNCE [181] is first proposed to approximate MI Equation (A.3.1):

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{2} \mathbb{E} \left[\log \frac{\exp(f_x(\mathbf{x}, \mathbf{y}))}{\exp(f_x(\mathbf{x}, \mathbf{y})) + \sum_j \exp(f_x(\mathbf{x}^j, \mathbf{y}))} + \log \frac{\exp(f_y(\mathbf{y}, \mathbf{x}))}{\exp(f_y(\mathbf{y}, \mathbf{x})) + \sum_j \exp(f_y(\mathbf{y}^j, \mathbf{x}))} \right], \quad (\text{A.4.1})$$

where $\mathbf{x}^j, \mathbf{y}^j$ are randomly sampled 2D and 3D views regarding to the anchored pair (\mathbf{x}, \mathbf{y}) . $f_x(\mathbf{x}, \mathbf{y}), f_y(\mathbf{y}, \mathbf{x})$ are scoring functions for the two corresponding views, whose formulation can be quite flexible. Here we use $f_x(\mathbf{x}, \mathbf{y}) = f_y(\mathbf{y}, \mathbf{x}) = \exp(\langle h_x, h_y \rangle)$.

Derivation of InfoNCE.

$$\begin{aligned} I(X; Y) - \log(K) &= \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} \left[\log \frac{1}{K} \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})} \right] \\ &= \sum_{\mathbf{x}^i, \mathbf{y}^i} \left[\log \frac{1}{K} \frac{p(\mathbf{x}^i, \mathbf{y}^i)}{p(\mathbf{x}^i)p(\mathbf{y}^i)} \right] \\ &\geq - \sum_{\mathbf{x}^i, \mathbf{y}^i} \left[\log \left(1 + (K-1) \frac{p(\mathbf{x}^i)p(\mathbf{y}^i)}{p(\mathbf{x}^i, \mathbf{y}^i)} \right) \right] \\ &= - \sum_{\mathbf{x}^i, \mathbf{y}^i} \left[\log \frac{\frac{p(\mathbf{x}^i, \mathbf{y}^i)}{p(\mathbf{x}^i)p(\mathbf{y}^i)} + (K-1)}{\frac{p(\mathbf{x}^i, \mathbf{y}^i)}{p(\mathbf{x}^i)p(\mathbf{y}^i)}} \right] \quad (\text{A.4.2}) \\ &\approx - \sum_{\mathbf{x}^i, \mathbf{y}^i} \left[\log \frac{\frac{p(\mathbf{x}^i, \mathbf{y}^i)}{p(\mathbf{x}^i)p(\mathbf{y}^i)} + (K-1) \mathbb{E}_{\mathbf{x}^j \neq \mathbf{x}^i} \frac{p(\mathbf{x}^j, \mathbf{y}^i)}{p(\mathbf{x}^j)p(\mathbf{y}^i)}}{\frac{p(\mathbf{x}^i, \mathbf{y}^i)}{p(\mathbf{x}^i)p(\mathbf{y}^i)}} \right] \quad // \textcircled{1} \\ &= \sum_{\mathbf{x}^i, \mathbf{y}^i} \left[\log \frac{\exp(f_x(\mathbf{x}^i, \mathbf{y}^i))}{\exp(f_x(\mathbf{x}^i, \mathbf{y}^i)) + \sum_{j=1}^K f_x(\mathbf{x}^j, \mathbf{y}^i)} \right], \end{aligned}$$

where we set $f_x(\mathbf{x}^i, \mathbf{y}^i) = \log \frac{p(\mathbf{x}^i, \mathbf{y}^i)}{p(\mathbf{x}^i)p(\mathbf{y}^i)}$.

Notice that in ①, we are using data $x \in X$ as the anchor points. If we use the $y \in Y$ as the anchor points and follow the similar steps, we can obtain

$$I(X; Y) - \log(K) \geq \sum_{\mathbf{y}^i, \mathbf{x}^i} \left[\log \frac{\exp(f_{\mathbf{y}}(\mathbf{y}^i, \mathbf{x}^i))}{\exp f_{\mathbf{y}}(\mathbf{y}^i, \mathbf{x}^i) + \sum_{j=1}^K \exp(f_{\mathbf{y}}(\mathbf{y}^j, \mathbf{x}^i))} \right]. \quad (\text{A.4.3})$$

Thus, by add both together, we can have the objective function as Equation (A.4.1).

A.4.2. EBM-NCE

We here provide an alternative approach to maximizing MI using energy-based model (EBM). To our best knowledge, we are the **first** to give the rigorous proof of using EBM to maximize the MI.

A.4.2.1. Energy-Based Model (EBM). Energy-based model (EBM) is a powerful tool for modeling the data distribution. The classic formulation is:

$$p(\mathbf{x}) = \frac{\exp(-E(\mathbf{x}))}{A}, \quad (\text{A.4.4})$$

where the bottleneck is the intractable partition function $A = \int_{\mathbf{x}} \exp(-E(\mathbf{x})) d\mathbf{x}$. Recently, there have been quite a lot progress along this direction [50, 79, 220, 221]. Noise Contrastive Estimation (NCE) [79] is one of the powerful tools here, as we will introduce later.

A.4.2.2. EBM for MI. Recall that our objective function is Equation (A.3.3): $\mathcal{L}_{\text{MI}} = \frac{1}{2}[H(Y|X) + H(X|Y)]$. Then we model the conditional likelihood with energy-based model (EBM). This gives us

$$\mathcal{L}_{\text{EBM}} = -\frac{1}{2} \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} \left[\log \frac{\exp(f_{\mathbf{x}}(\mathbf{x}, \mathbf{y}))}{A_{\mathbf{x}|\mathbf{y}}} + \log \frac{\exp(f_{\mathbf{y}}(\mathbf{y}, \mathbf{x}))}{A_{\mathbf{y}|\mathbf{x}}} \right], \quad (\text{A.4.5})$$

where $f_{\mathbf{x}}(\mathbf{x}, \mathbf{y}) = -E(\mathbf{x}|\mathbf{y})$ and $f_{\mathbf{y}}(\mathbf{y}, \mathbf{x}) = -E(\mathbf{y}|\mathbf{x})$ are the negative energy functions, and $A_{\mathbf{x}|\mathbf{y}}$ and $A_{\mathbf{y}|\mathbf{x}}$ are the corresponding partition functions.

Under the EBM framework, if we solve Equation (A.4.5) with Noise Contrastive Estimation (NCE) [79], the final EBM-NCE objective is

$$\begin{aligned} \mathcal{L}_{\text{EBM-NCE}} = & -\frac{1}{2} \mathbb{E}_{p_{\text{data}}(\mathbf{y})} \left[\mathbb{E}_{p_n(\mathbf{x}|\mathbf{y})} [\log (1 - \sigma(f_{\mathbf{x}}(\mathbf{x}, \mathbf{y})))] + \mathbb{E}_{p_{\text{data}}(\mathbf{x}|\mathbf{y})} [\log \sigma(f_{\mathbf{x}}(\mathbf{x}, \mathbf{y}))] \right] \\ & -\frac{1}{2} \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[\mathbb{E}_{p_n(\mathbf{y}|\mathbf{x})} [\log (1 - \sigma(f_{\mathbf{y}}(\mathbf{y}, \mathbf{x})))] + \mathbb{E}_{p_{\text{data}}(\mathbf{y}|\mathbf{x})} [\log \sigma(f_{\mathbf{y}}(\mathbf{y}, \mathbf{x}))] \right]. \end{aligned} \quad (\text{A.4.6})$$

Next we will give the detailed derivations.

A.4.2.3. Derivation of conditional EBM with NCE. WLOG, let's consider the $p_{\theta}(\mathbf{x}|\mathbf{y})$ first, and by EBM it is as follows:

$$p_{\theta}(\mathbf{x}|\mathbf{y}) = \frac{\exp(-E(\mathbf{x}|\mathbf{y}))}{\int \exp(-E(\tilde{\mathbf{x}}|\mathbf{y})) d\tilde{\mathbf{x}}} = \frac{\exp(f_{\mathbf{x}}(\mathbf{x}, \mathbf{y}))}{\int \exp(f_{\mathbf{x}}(\tilde{\mathbf{x}}|\mathbf{y})) d\tilde{\mathbf{x}}} = \frac{\exp(f_{\mathbf{x}}(\mathbf{x}, \mathbf{y}))}{A_{\mathbf{x}|\mathbf{y}}}. \quad (\text{A.4.7})$$

Then we solve this using NCE. NCE handles the intractability issue by transforming it as a binary classification task. We take the partition function $A_{\mathbf{x}|\mathbf{y}}$ as a parameter, and introduce a noise distribution p_n . Based on this, we introduce a mixture model: $\mathbf{z} = 0$ if the conditional $\mathbf{x}|\mathbf{y}$ is from $p_n(\mathbf{x}|\mathbf{y})$, and $\mathbf{z} = 1$ if $\mathbf{x}|\mathbf{y}$ is from $p_{\text{data}}(\mathbf{x}|\mathbf{y})$. So the joint distribution is:

$$p_{n,\text{data}}(\mathbf{x}|\mathbf{y}) = p(\mathbf{z} = 1)p_{\text{data}}(\mathbf{x}|\mathbf{y}) + p(\mathbf{z} = 0)p_n(\mathbf{x}|\mathbf{y})$$

The posterior of $p(\mathbf{z} = 0|\mathbf{x},\mathbf{y})$ is

$$p_{n,\text{data}}(\mathbf{z} = 0|\mathbf{x},\mathbf{y}) = \frac{p(\mathbf{z} = 0)p_n(\mathbf{x}|\mathbf{y})}{p(\mathbf{z} = 0)p_n(\mathbf{x}|\mathbf{y}) + p(\mathbf{z} = 1)p_{\text{data}}(\mathbf{x}|\mathbf{y})} = \frac{\nu \cdot p_n(\mathbf{x}|\mathbf{y})}{\nu \cdot p_n(\mathbf{x}|\mathbf{y}) + p_{\text{data}}(\mathbf{x}|\mathbf{y})},$$

where $\nu = \frac{p(\mathbf{z}=0)}{p(\mathbf{z}=1)}$.

Similarly, we can have the joint distribution under EBM framework as:

$$p_{n,\theta}(\mathbf{x}) = p(\mathbf{z} = 0)p_n(\mathbf{x}|\mathbf{y}) + p(\mathbf{z} = 1)p_\theta(\mathbf{x}|\mathbf{y})$$

And the corresponding posterior is:

$$p_{n,\theta}(\mathbf{z} = 0|\mathbf{x},\mathbf{y}) = \frac{p(\mathbf{z} = 0)p_n(\mathbf{x}|\mathbf{y})}{p(\mathbf{z} = 0)p_n(\mathbf{x}|\mathbf{y}) + p(\mathbf{z} = 1)p_\theta(\mathbf{x}|\mathbf{y})} = \frac{\nu \cdot p_n(\mathbf{x}|\mathbf{y})}{\nu \cdot p_n(\mathbf{x}|\mathbf{y}) + p_\theta(\mathbf{x}|\mathbf{y})}$$

We indirectly match $p_\theta(\mathbf{x}|\mathbf{y})$ to $p_{\text{data}}(\mathbf{x}|\mathbf{y})$ by fitting $p_{n,\theta}(\mathbf{z}|\mathbf{x},\mathbf{y})$ to $p_{n,\text{data}}(\mathbf{z}|\mathbf{x},\mathbf{y})$ by minimizing their KL-divergence:

$$\begin{aligned} & \min_{\theta} D_{\text{KL}}(p_{n,\text{data}}(\mathbf{z}|\mathbf{x},\mathbf{y}) || p_{n,\theta}(\mathbf{z}|\mathbf{x},\mathbf{y})) \\ &= \mathbb{E}_{p_{n,\text{data}}(\mathbf{x},\mathbf{z}|\mathbf{y})} [\log p_{n,\theta}(\mathbf{z}|\mathbf{x},\mathbf{y})] \\ &= \int \sum_{\mathbf{z}} p_{n,\text{data}}(\mathbf{x},\mathbf{z}|\mathbf{y}) \cdot \log p_{n,\theta}(\mathbf{z}|\mathbf{x},\mathbf{y}) d\mathbf{x} \\ &= \int \left\{ p(\mathbf{z} = 0)p_{n,\text{data}}(\mathbf{x}|\mathbf{y},\mathbf{z} = 0) \log p_{n,\theta}(\mathbf{z} = 0|\mathbf{x},\mathbf{y}) \right. \\ & \quad \left. + p(\mathbf{z} = 1)p_{n,\text{data}}(\mathbf{x}|\mathbf{y},\mathbf{z} = 1) \log p_{n,\theta}(\mathbf{z} = 1|\mathbf{x},\mathbf{y}) \right\} d\mathbf{x} \\ &= \nu \cdot \mathbb{E}_{p_n(\mathbf{x}|\mathbf{y})} \left[\log p_{n,\theta}(\mathbf{z} = 0|\mathbf{x},\mathbf{y}) \right] + \mathbb{E}_{p_{\text{data}}(\mathbf{x}|\mathbf{y})} \left[\log p_{n,\theta}(\mathbf{z} = 1|\mathbf{x},\mathbf{y}) \right] \\ &= \nu \cdot \mathbb{E}_{p_n(\mathbf{x}|\mathbf{y})} \left[\log \frac{\nu \cdot p_n(\mathbf{x}|\mathbf{y})}{\nu \cdot p_n(\mathbf{x}|\mathbf{y}) + p_\theta(\mathbf{x}|\mathbf{y})} \right] + \mathbb{E}_{p_{\text{data}}(\mathbf{x}|\mathbf{y})} \left[\log \frac{p_\theta(\mathbf{x}|\mathbf{y})}{\nu \cdot p_n(\mathbf{x}|\mathbf{y}) + p_\theta(\mathbf{x}|\mathbf{y})} \right]. \end{aligned} \tag{A.4.8}$$

This optimal distribution is an estimation to the actual distribution (or data distribution), *i.e.*, $p_\theta(\mathbf{x}|\mathbf{y}) \approx p_{\text{data}}(\mathbf{x}|\mathbf{y})$. We can follow the similar steps for $p_\theta(\mathbf{y}|\mathbf{x}) \approx p_{\text{data}}(\mathbf{y}|\mathbf{x})$. Thus following Equation (A.4.8), the objective function is to maximize

$$\nu \cdot \mathbb{E}_{p_{\text{data}}(\mathbf{y})} \mathbb{E}_{p_n(\mathbf{x}|\mathbf{y})} \left[\log \frac{\nu \cdot p_n(\mathbf{x}|\mathbf{y})}{\nu \cdot p_n(\mathbf{x}|\mathbf{y}) + p_\theta(\mathbf{x}|\mathbf{y})} \right] + \mathbb{E}_{p_{\text{data}}(\mathbf{y})} \mathbb{E}_{p_{\text{data}}(\mathbf{x}|\mathbf{y})} \left[\log \frac{p_\theta(\mathbf{x}|\mathbf{y})}{\nu \cdot p_n(\mathbf{x}|\mathbf{y}) + p_\theta(\mathbf{x}|\mathbf{y})} \right]. \tag{A.4.9}$$

The we will adopt three strategies to approximate Equation (A.4.9):

- (1) **Self-normalization.** When the EBM is very expressive, *i.e.*, using deep neural network for modeling, we can assume it is able to approximate the normalized density directly [172, 220]. In other words, we can set the partition function $A = 1$. This is a self-normalized EBM-NCE, with normalizing constant close to 1, *i.e.*, $p(\mathbf{x}) = \exp(-E(\mathbf{x})) = \exp(f(\mathbf{x}))$ in Equation (A.4.4).
- (2) **Exponential tilting term.** Exponential tilting term [4] is another useful trick. It models the distribution as $\tilde{p}_\theta(\mathbf{x}) = q(\mathbf{x}) \exp(-E_\theta(\mathbf{x}))$, where $q(\mathbf{x})$ is the reference distribution. If we use the same reference distribution as the noise distribution, the tilted probability is $\tilde{p}_\theta(\mathbf{x}) = p_n(\mathbf{x}) \exp(-E_\theta(\mathbf{x}))$ in Equation (A.4.4).
- (3) **Sampling.** For many cases, we only need to sample 1 negative points for each data, *i.e.*, $\nu = 1$.

Following these three disciplines, the objective function to optimize $p_\theta(\mathbf{x}|\mathbf{y})$ becomes

$$\begin{aligned}
& \mathbb{E}_{p_n(\mathbf{x}|\mathbf{y})} \left[\log \frac{p_n(\mathbf{x}|\mathbf{y})}{p_n(\mathbf{x}|\mathbf{y}) + \tilde{p}_\theta(\mathbf{x}|\mathbf{y})} \right] + \mathbb{E}_{p_{\text{data}}(\mathbf{x}|\mathbf{y})} \left[\log \frac{\tilde{p}_\theta(\mathbf{x}|\mathbf{y})}{p_n(\mathbf{x}|\mathbf{y}) + \tilde{p}_\theta(\mathbf{x}|\mathbf{y})} \right] \\
&= \mathbb{E}_{p_n(\mathbf{x}|\mathbf{y})} \left[\log \frac{1}{1 + p_\theta(\mathbf{x}|\mathbf{y})} \right] + \mathbb{E}_{p_{\text{data}}(\mathbf{x}|\mathbf{y})} \left[\log \frac{p_\theta(\mathbf{x}|\mathbf{y})}{1 + p_\theta(\mathbf{x}|\mathbf{y})} \right] \\
&= \mathbb{E}_{p_n(\mathbf{x}|\mathbf{y})} \left[\log \frac{\exp(-f_x(\mathbf{x}, \mathbf{y}))}{\exp(-f_x(\mathbf{x}, \mathbf{y})) + 1} \right] + \mathbb{E}_{p_{\text{data}}(\mathbf{x}|\mathbf{y})} \left[\log \frac{1}{\exp(-f_x(\mathbf{x}, \mathbf{y})) + 1} \right] \\
&= \mathbb{E}_{p_n(\mathbf{x}|\mathbf{y})} \left[\log \left(1 - \sigma(f_x(\mathbf{x}, \mathbf{y})) \right) \right] + \mathbb{E}_{p_{\text{data}}(\mathbf{x}|\mathbf{y})} \left[\log \sigma(f_x(\mathbf{x}, \mathbf{y})) \right].
\end{aligned} \tag{A.4.10}$$

Thus, the final EBM-NCE contrastive SSL objective is

$$\begin{aligned}
\mathcal{L}_{\text{EBM-NCE}} &= -\frac{1}{2} \mathbb{E}_{p_{\text{data}}(\mathbf{y})} \left[\mathbb{E}_{p_n(\mathbf{x}|\mathbf{y})} \log \left(1 - \sigma(f_x(\mathbf{x}, \mathbf{y})) \right) + \mathbb{E}_{p_{\text{data}}(\mathbf{x}|\mathbf{y})} \log \sigma(f_x(\mathbf{x}, \mathbf{y})) \right] \\
&\quad - \frac{1}{2} \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[\mathbb{E}_{p_n(\mathbf{y}|\mathbf{x})} \log \left(1 - \sigma(f_y(\mathbf{y}, \mathbf{x})) \right) + \mathbb{E}_{p_{\text{data}}(\mathbf{y}|\mathbf{x})} \log \sigma(f_y(\mathbf{y}, \mathbf{x})) \right].
\end{aligned} \tag{A.4.11}$$

A.4.3. EBM-NCE v.s. JSE and InfoNCE

We acknowledge that there are many other contrastive objectives [188] that can be used to maximize MI. However, in the research line of graph SSL, as summarized in several recent survey papers [160, 262, 265], the two most used ones are InfoNCE and Jensen-Shannon Estimator (JSE) [91, 180].

We conclude that JSE is very similar to EBM-NCE, while the underlying perspectives are totally different, as explained below.

- (1) **Derivation and Intuition.** Derivation process and underlying intuition are different. JSE [180] starts from f-divergence, then with variational estimation and Fenchel duality on function f . Our proposed EBM-NCE is more straightforward: it models

the conditional distribution in the MI lower bound Equation (A.3.3) with EBM, and solves it using NCE.

- (2) **Flexibility.** Modeling the conditional distribution with EBM provides a broader family of algorithms. NCE is just one solution to it, and recent progress on score matching [220, 221] and contrastive divergence [50], though no longer contrastive SSL, adds on more promising directions. Thus, EBM can provide a potential unified framework for structuring our understanding of self-supervised learning.
- (3) **Noise distribution.** Starting from [91], all the following works on graph SSL [160, 227, 262, 265] have been adopting the empirical distribution for noise distribution. However, this is not the case in EBM-NCE. Classic EBM-NCE uses fixed distribution, while more recent work [4] extends it with adaptively learnable noise distribution. With this discipline, more advanced sampling strategies (w.r.t. the noise distribution) can be proposed, *e.g.*, adversarial negative sampling in [95].

In the above, we conclude three key differences between EBM-NCE and JSE, plus the solid and straightforward derivations on EBM-NCE. We believe this can provide a insightful perspective of SSL to the community.

According to the empirical results Section 4.4, we observe that EBM-NCE is better than InfoNCE. This can be explained using the claim from [120], where the main technical contribution is to construct many positives and many negatives per anchor point. The binary cross-entropy in EBM-NCE is able to realize this to some extent: make all the positive pairs positive and all the negative pairs negative, where the softmax-based cross-entropy fails to capture this, as in InfoNCE.

To conclude, we are introduce using EBM in modeling MI, which opens many potential venues. As for contrastive SSL, EBM-NCE provides a better perspective than JSE, and is better than InfoNCE on graph-level self-supervised learning.

A.5. Generative Self-Supervised Learning

Generative SSL is another classic track for unsupervised pre-training [122, 123, 131], though the main focus is on distribution learning. In GraphMVP, we start with VAE for the following reasons:

- (1) One of the biggest attributes of our problem is that the mapping between two views are stochastic: multiple 3D conformers can correspond to the same 2D topology. Thus, we expect a stochastic model [178] like VAE, instead of the deterministic ones.
- (2) For pre-training and fine-tuning, we need to learn an explicit and powerful representation function that can be used for downstream tasks.
- (3) The decoder for structured data like graph are often complicated, *e.g.*, the auto-regressive generation. This makes them suboptimal.

To cope with these challenges, in GraphMVP, we start with VAE-like generation model, and later propose a *light-weighted* and *smart* surrogate loss as objective function. Notice that for notation simplicity, for this section, we use h_y and h_x to delegate the 2D and 3D GNN respectively.

A.5.1. Variational Molecule Reconstruction

As shown in Equation (A.3.3), our main motivation is to model the conditional likelihood:

$$\mathcal{L}_{\text{MI}} = -\frac{1}{2}\mathbb{E}_{p(x,y)}[\log p(\mathbf{x}|\mathbf{y}) + \log p(\mathbf{y}|\mathbf{x})]$$

By introducing a reparameterized variable $\mathbf{z}_x = \mu_x + \sigma_x \odot \epsilon$, where μ_x and σ_x are two flexible functions on h_x , $\epsilon \sim \mathcal{N}(0, I)$ and \odot is the element-wise production, we have a lower bound on the conditional likelihood:

$$\log p(\mathbf{y}|\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}_x|\mathbf{x})}[\log p(\mathbf{y}|\mathbf{z}_x)] - KL(q(\mathbf{z}_x|\mathbf{x})||p(\mathbf{z}_x)). \quad (\text{A.5.1})$$

Similarly, we have

$$\log p(\mathbf{x}|\mathbf{y}) \geq \mathbb{E}_{q(\mathbf{z}_y|\mathbf{y})}[\log p(\mathbf{x}|\mathbf{z}_y)] - KL(q(\mathbf{z}_y|\mathbf{y})||p(\mathbf{z}_y)), \quad (\text{A.5.2})$$

where $\mathbf{z}_y = \mu_y + \sigma_y \odot \epsilon$. Here μ_y and σ_y are flexible functions on h_y , and $\epsilon \sim \mathcal{N}(0, I)$. For implementation, we take multi-layer perceptrons (MLPs) for $\mu_x, \mu_y, \sigma_x, \sigma_y$.

Both the above objectives are composed of a conditional log-likelihood and a KL-divergence. The conditional log-likelihood has also been recognized as the *reconstruction term*: it is essentially to reconstruct the 3D conformers (\mathbf{y}) from the sampled 2D molecular graph representation (\mathbf{z}_x). However, performing the graph reconstruction on the data space is not easy: since molecules are discrete, modeling and measuring are not trivial.

A.5.2. Variational Representation Reconstruction

To cope with data reconstruction issue, we propose a novel generative loss termed variation representation reconstruction (VRR). The pipeline is in Figure 16.

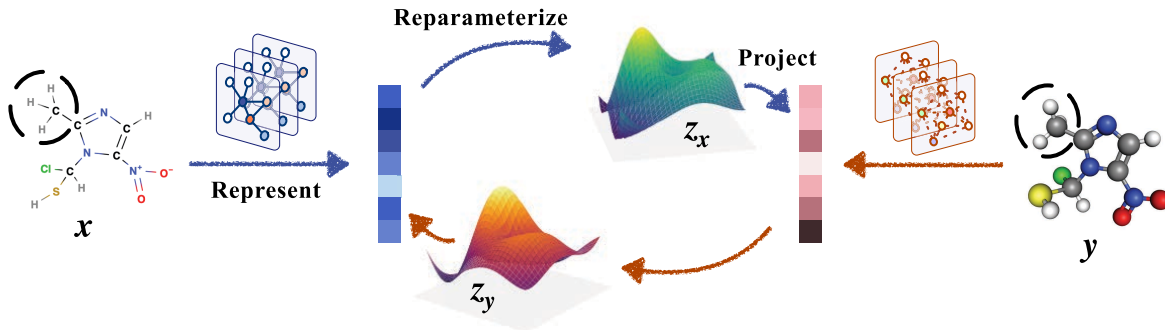


Figure 16. VRR SSL in GraphMVP. The black dashed circles represent subgraph masking.

Our proposed solution is very straightforward. Recall that MI is invariant to continuous bijective function [12]. So suppose we have a representation function h_y satisfying this condition, and this can guide us a surrogate loss by transferring the reconstruction from data space to the continuous representation space:

$$\mathbb{E}_{q(z_x|x)}[\log p(\mathbf{y}|z_x)] = -\mathbb{E}_{q(z_x|x)}[\|h_y(g_x(z_x)) - h_y(\mathbf{y})\|_2^2] + C,$$

where g_x is the decoder and C is a constant, and this introduces to using the mean-squared error (MSE) for **reconstruction on the representation space**.

Then for the reconstruction, current formula has two steps: i) the latent code z_x is first mapped to molecule space, and ii) it is mapped to the representation space. We can approximate these two mappings with one projection step, by directly projecting the latent code z_x to the 3D representation space, *i.e.*, $q_x(z_x) \approx h_y(g_x(z_x))$. This gives us a variation representation reconstruction (VRR) SSL objective as below:

$$\mathbb{E}_{q(z_x|x)}[\log p(\mathbf{y}|z_x)] = -\mathbb{E}_{q(z_x|x)}[\|q_x(z_x) - h_y(\mathbf{y})\|_2^2] + C.$$

β -VAE. We consider introducing a β variable [88] to control the disentanglement of the latent representation. To be more specific, we would have

$$\log p(\mathbf{y}|x) \geq \mathbb{E}_{q(z_x|x)}[\log p(\mathbf{y}|z_x)] - \beta \cdot KL(q(z_x|x)||p(z_x)). \quad (\text{A.5.3})$$

Stop-gradient. For the optimization on variational representation reconstruction, related work have found that adding the stop-gradient operator (SG) as a regularizer can make the training more stable without collapse both empirically [30, 75] and theoretically [235].

Here, we may as well utilize this SG operation in the objective function:

$$\mathbb{E}_{q(\mathbf{z}_x|\mathbf{x})}[\log p(\mathbf{y}|\mathbf{z}_x)] = -\mathbb{E}_{q(\mathbf{z}_x|\mathbf{x})}[\|q_x(\mathbf{z}_x) - \text{SG}(h_y(\mathbf{y}))\|_2^2] + C. \quad (\text{A.5.4})$$

Objective function for VRR. Thus, combining both two regularizers mentioned above, the final objective function for VRR is:

$$\begin{aligned} \mathcal{L}_{\text{VRR}} = & \frac{1}{2} \left[\mathbb{E}_{q(\mathbf{z}_x|\mathbf{x})}[\|q_x(\mathbf{z}_x) - \text{SG}(h_y)\|_2^2] + \mathbb{E}_{q(\mathbf{z}_y|\mathbf{y})}[\|q_y(\mathbf{z}_y) - \text{SG}(h_x)\|_2^2] \right] \\ & + \frac{\beta}{2} \cdot \left[KL(q(\mathbf{z}_x|\mathbf{x})||p(\mathbf{z}_x)) + KL(q(\mathbf{z}_y|\mathbf{y})||p(\mathbf{z}_y)) \right]. \end{aligned} \quad (\text{A.5.5})$$

Note that MI is invariant to continuous bijective function [12], thus this surrogate loss would be exact if the encoding function h_y and h_x satisfy this condition. However, we find GNN (both GIN and SchNet) can, though do not meet the condition, provide quite robust performance empirically, which justify the effectiveness of VRR.

A.5.3. Variational Representation Reconstruction and Non-Contrastive SSL

By introducing VRR, we provide another perspective to understand the generative SSL, including the recently-proposed non-contrastive SSL [30, 75].

We provide a unified structure on the intra-data generative SSL:

- Reconstruction to the data space, like Equations (3.3), (A.5.1) and (A.5.2).
- Reconstruction to the representation space, *i.e.*, VRR in Equation (A.5.5).
 - If we **remove the stochasticity**, then it is simply the representation reconstruction (RR), as we tested in the ablation study Section 4.4.
 - If we **remove the stochasticity** and assume two views are **sharing the same representation function**, like CNN for multi-view learning on images, then it is reduced to the BYOL [75] and SimSiam [30]. In other words, these recently-proposed non-contrastive SSL methods are indeed special cases of VRR.

A.6. Dataset Overview

A.6.1. Pre-Training Dataset Overview

In this section, we provide the basic statistics of the pre-training dataset (GEOM).

In Figure 17, we plot the histogram (logarithm scale on the y-axis) and cumulative distribution on the number of conformers of each molecule. As shown by the histogram and curves, there are certain number of molecules having over 1000 possible 3d conformer structures, while over 80% of molecules have less than 100 conformers.

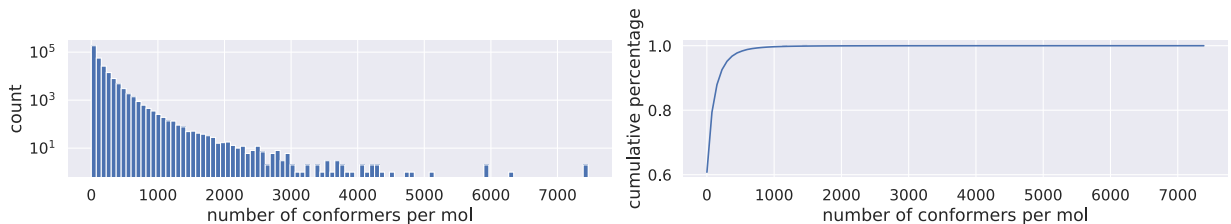


Figure 17. Statistics on the conformers of each molecule

In Figure 17, we plot the histogram of the summation of top (descending sorted by weights) $\{1,5,10,20\}$ conformer weights. The physical meaning of the weight is the portion of each conformer occurred in nature. We observe that the top 5 or 10 conformers are sufficient as they have dominated nearly all the natural observations. Such long-tailed distribution is also in alignment with our findings in the ablation studies. We find that utilizing top five conformers in the GraphMVP has reached an idealised spot between effectiveness and efficiency.

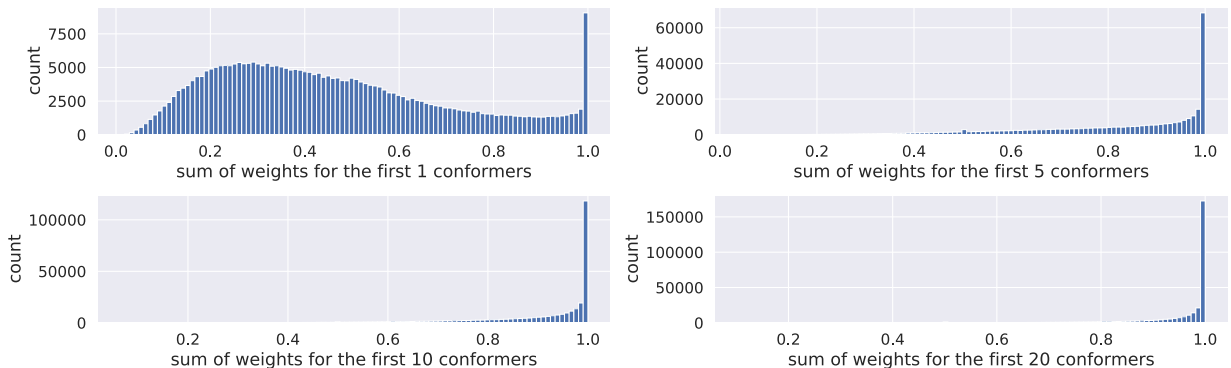


Figure 18. Sum of occurrence weights for the top major conformers

A.6.2. Downstream Dataset Overview

In this section, we review the four main categories of datasets used for downstream tasks. **Molecular Property: Pharmacology.** The Blood-Brain Barrier Penetration (BBBP) [165] dataset measures whether a molecule will penetrate the central nervous system. All three datasets, Tox21 [237], ToxCast [263], and ClinTox [66] are related to the toxicity of molecular compounds. The Side Effect Resource (SIDER) [128] dataset stores the adverse drug reactions on a marketed drug database.

Molecular Property: Physical Chemistry. Dataset proposed in [42] measures aqueous solubility of the molecular compounds. Lipophilicity (Lipo) dataset is a subset of ChEMBL [65] measuring the molecule octanol/water distribution coefficient. CEP dataset is a subset of the Havard Clean Energy Project (CEP) [80], which estimates the organic photovoltaic efficiency. **Molecular Property: Biophysics.** Maximum Unbiased Validation (MUV) [199] is another sub-database from PCBA, and is obtained by applying a refined nearest neighbor analysis. HIV is from the Drug Therapeutics Program (DTP) AIDS Antiviral Screen [277], and it aims at predicting inhibit HIV replication. BACE measures the binding results for a set of inhibitors of β -secretase 1 (BACE-1), and is gathered in MoleculeNet [263]. Malaria [62] measures the drug efficacy against the parasite that causes malaria.

Drug-Target Affinity. Davis [41] measures the binding affinities between kinase inhibitors and kinases, scored by the K_d value (kinase dissociation constant). KIBA [233] contains binding affinities for kinase inhibitors from different sources, including K_i , K_d and IC_{50} . KIBA scores [182] are constructed to optimize the consistency among these values.

Table 21. Summary for the molecule chemical datasets.

Dataset	Task	# Tasks	# Molecules	# Proteins	# Molecule-Protein pairs
BBBP	Classification	1	2,039	-	-
Tox21	Classification	12	7,831	-	-
ToxCast	Classification	617	8,576	-	-
Sider	Classification	27	1,427	-	-
ClinTox	Classification	2	1,478	-	-
MUV	Classification	17	93,087	-	-
HIV	Classification	1	41,127	-	-
Bace	Classification	1	1,513	-	-
Delaney	Regression	1	1,128	-	-
Lipo	Regression	1	4,200	-	-
Malaria	Regression	1	9,999	-	-
CEP	Regression	1	29,978	-	-
Davis	Regression	1	68	379	30,056
KIBA	Regression	1	2,068	229	118,254

A.7. Experiments Details

A.7.1. Self-supervised Learning Baselines

For the SSL baselines in main results (Table 1), generally we can match with the original paper, even though most of them are using larger pre-training datasets, like ZINC-2m. Yet, we would like to add some specifications.

- G- $\{\text{Contextual, Motif}\}$ [200] proposes a new GNN model for backbone model, and does pre-training on a larger dataset. Both settings are different from us.
- JOAO [273] has two versions in the original paper. In this paper, we run both versions and report the optimal one.
- Almost all the graph SSL baselines are reporting the test performance with optimal validation error, while GraphLoG [267] reports 73.2 in the paper with the last-epoch performance. This can be over-optimized in terms of overfitting, and here we rerun it with the same downstream evaluation strategy as a fair comparison.

A.7.2. Ablation Study: The Effect of Masking Ratio and Number of Conformers

Table 22. Full results for ablation of masking ratio M ($C = 0.15$), MVP is short for GraphMVP.

	M	BBBP	Tox21	ToxCast	Sider	ClinTox	MUV	HIV	Bace	Avg
–	–	65.4(2.4)	74.9(0.8)	61.6(1.2)	58.0(2.4)	58.8(5.5)	71.0(2.5)	75.3(0.5)	72.6(4.9)	67.21
MVP	0	69.4 (1.0)	75.3 (0.5)	62.8 (0.2)	61.9 (0.5)	74.4 (1.3)	74.6 (1.4)	74.6 (1.0)	76.0 (2.0)	71.12
	0.15	68.5 (0.2)	74.5 (0.4)	62.7 (0.1)	62.3 (1.6)	79.0 (2.5)	75.0 (1.4)	74.8 (1.4)	76.8 (1.1)	71.69
	0.3	68.6 (0.3)	74.9 (0.6)	62.8 (0.4)	60.0 (0.6)	74.8 (7.8)	74.7 (0.8)	75.5 (1.1)	82.9 (1.7)	71.79
MVP-G	0	72.4 (1.3)	74.7 (0.6)	62.4 (0.2)	60.3 (0.7)	76.2 (5.7)	76.6 (1.7)	76.4 (1.7)	78.0 (1.1)	72.15
	0.15	70.8 (0.5)	75.9 (0.5)	63.1 (0.2)	60.2 (1.1)	79.1 (2.8)	77.7 (0.6)	76.0 (0.1)	79.3 (1.5)	72.76
	0.3	69.5 (0.5)	74.6 (0.6)	62.7 (0.3)	60.8 (1.2)	80.7 (2.0)	77.8 (2.5)	76.2 (0.5)	81.0 (1.0)	72.91
MVP-C	0	71.5 (0.9)	75.4 (0.3)	63.6 (0.5)	61.8 (0.6)	77.3 (1.2)	75.8 (0.6)	76.1 (0.9)	79.8 (0.4)	72.66
	0.15	72.4 (1.6)	74.4 (0.2)	63.1 (0.4)	63.9 (1.2)	77.5 (4.2)	75.0 (1.0)	77.0 (1.2)	81.2 (0.9)	73.07
	0.3	70.7 (0.8)	74.6 (0.3)	63.8 (0.7)	60.4 (0.6)	83.5 (3.2)	74.2 (1.6)	76.0 (1.0)	82.2 (2.2)	73.17

Table 23. Full results for ablation of # conformers C ($M = 0.5$), MVP is short for GraphMVP.

	C	BBBP	Tox21	ToxCast	Sider	ClinTox	MUV	HIV	Bace	Avg
–	–	65.4(2.4)	74.9(0.8)	61.6(1.2)	58.0(2.4)	58.8(5.5)	71.0(2.5)	75.3(0.5)	72.6(4.9)	67.21
MVP	1	69.2 (1.0)	74.7 (0.4)	62.5 (0.2)	63.0 (0.4)	73.9 (7.2)	76.2 (0.4)	75.3 (1.1)	78.0 (0.5)	71.61
	5	68.5 (0.2)	74.5 (0.4)	62.7 (0.1)	62.3 (1.6)	79.0 (2.5)	75.0 (1.4)	74.8 (1.4)	76.8 (1.1)	71.69
	10	68.3 (0.5)	74.2 (0.6)	63.2 (0.5)	61.4 (1.0)	80.6 (0.8)	75.4 (2.4)	75.5 (0.6)	79.1 (2.3)	72.20
	20	68.7 (0.5)	74.9 (0.3)	62.7 (0.3)	60.8 (0.7)	75.8 (0.5)	76.3 (1.5)	77.4 (0.3)	82.3 (0.8)	72.39
MVP-G	1	70.9 (0.4)	75.3 (0.7)	62.8 (0.5)	61.2 (0.6)	81.4 (3.7)	74.2 (2.1)	76.4 (0.6)	80.2 (0.7)	72.80
	5	70.8 (0.5)	75.9 (0.5)	63.1 (0.2)	60.2 (1.1)	79.1 (2.8)	77.7 (0.6)	76.0 (0.1)	79.3 (1.5)	72.76
	10	70.2 (0.9)	74.9 (0.4)	63.4 (0.4)	60.8 (1.0)	80.6 (0.4)	76.4 (2.0)	77.0 (0.3)	77.4 (1.3)	72.59
	20	69.5 (0.4)	74.9 (0.4)	63.3 (0.1)	60.8 (0.3)	81.2 (0.5)	77.3 (2.7)	76.9 (0.3)	80.1 (0.5)	73.00
MVP-C	1	69.7 (0.9)	74.9 (0.5)	64.1 (0.5)	61.0 (1.4)	78.3 (2.7)	75.7 (1.5)	74.7 (0.8)	81.3 (0.7)	72.46
	5	72.4 (1.6)	74.4 (0.2)	63.1 (0.4)	63.9 (1.2)	77.5 (4.2)	75.0 (1.0)	77.0 (1.2)	81.2 (0.9)	73.07
	10	69.5 (1.5)	74.5 (0.5)	63.9 (0.9)	60.9 (0.4)	81.1 (1.8)	76.8 (1.5)	76.0 (0.8)	82.0 (1.0)	73.09
	20	72.1 (0.4)	73.4 (0.7)	63.9 (0.3)	63.0 (0.7)	78.8 (2.4)	74.1 (1.0)	74.8 (0.9)	84.1 (0.6)	73.02

A.7.3. Ablation Study: Effect of Each Loss Component

Table 24. Molecular graph property prediction, we set $C=5$ and $M=0.15$ for GraphMVP methods.

	BBBP	Tox21	ToxCast	Sider	ClinTox	MUV	HIV	Bace	Avg
# Molecules	2,039	7,831	8,575	1,427	1,478	93,087	41,127	1,513	-
# Tasks	1	12	617	27	2	17	1	1	-
-	65.4(2.4)	74.9(0.8)	61.6(1.2)	58.0(2.4)	58.8(5.5)	71.0(2.5)	75.3(0.5)	72.6(4.9)	67.21
InfoNCE only	68.9(1.2)	74.2(0.3)	62.8(0.2)	59.7(0.7)	57.8(11.5)	73.6(1.8)	76.1(0.6)	77.6(0.3)	68.85
EBM-NCE only	68.0(0.3)	74.3(0.4)	62.6(0.3)	61.3(0.4)	66.0(6.0)	73.1(1.6)	76.4(1.0)	79.6(1.7)	70.15
VAE only	67.6(1.8)	73.2(0.5)	61.9(0.4)	60.5(0.2)	59.7(1.6)	78.6(0.7)	77.4(0.6)	75.4(2.1)	69.29
AE only	70.5(0.4)	75.0(0.4)	62.4(0.4)	61.0(1.4)	53.8(1.0)	74.1(2.9)	76.3(0.5)	77.9(0.9)	68.89
InfoNCE + VAE	69.6(1.1)	75.4(0.6)	63.2(0.3)	59.9(0.4)	69.3(14.0)	76.5(1.3)	76.3(0.2)	75.2(2.7)	70.67
EBM-NCE + VAE	68.5(0.2)	74.5(0.4)	62.7(0.1)	62.3(1.6)	79.0(2.5)	75.0(1.4)	74.8(1.4)	76.8(1.1)	71.69
InfoNCE + AE	65.1(3.1)	75.4(0.7)	62.5(0.5)	59.2(0.6)	77.2(1.8)	72.4(1.4)	75.8(0.6)	77.1(0.8)	70.60
EBM-NCE + AE	69.4(1.0)	75.2(0.1)	62.4(0.4)	61.5(0.9)	71.1(6.0)	73.3(0.3)	75.2(0.6)	79.3(1.1)	70.94

A.7.4. Broader Range of Downstream Tasks: Molecular Property Prediction Prediction

Table 25. Results for four molecular property prediction tasks (regression). For each downstream task, we report the mean (and standard variance) RMSE of 3 seeds with scaffold splitting. For GraphMVP, we set $M = 0.15$ and $C = 5$. The best performance for each task is marked in **bold**.

	ESOL	Lipo	Malaria	CEP	Avg
-	1.178 (0.044)	0.744 (0.007)	1.127 (0.003)	1.254 (0.030)	1.07559
AM	1.112 (0.048)	0.730 (0.004)	1.119 (0.014)	1.256 (0.000)	1.05419
CP	1.196 (0.037)	0.702 (0.020)	1.101 (0.015)	1.243 (0.025)	1.06059
JOAO	1.120 (0.019)	0.708 (0.007)	1.145 (0.010)	1.293 (0.003)	1.06631
GraphMVP	1.091 (0.021)	0.718 (0.016)	1.114 (0.013)	1.236 (0.023)	1.03968
GraphMVP-G	1.064 (0.045)	0.691 (0.013)	1.106 (0.013)	1.228 (0.001)	1.02214
GraphMVP-C	1.029 (0.033)	0.681 (0.010)	1.097 (0.017)	1.244 (0.009)	1.01283

A.7.5. Broader Range of Downstream Tasks: Drug-Target Affinity Prediction

Table 26. Results for two drug-target affinity prediction tasks (regression). For each downstream task, we report the mean (and standard variance) MSE of 3 seeds with random splitting. For GraphMVP, we set $M = 0.15$ and $C = 5$. The best performance for each task is marked in **bold**.

	Davis	KIBA	Avg
	0.286 (0.006)	0.206 (0.004)	0.24585
AM	0.291 (0.007)	0.203 (0.003)	0.24730
CP	0.279 (0.002)	0.198 (0.004)	0.23823
JOAO	0.281 (0.004)	0.196 (0.005)	0.23871
GraphMVP	0.280 (0.005)	0.178 (0.005)	0.22860
GraphMVP-G	0.274 (0.002)	0.175 (0.001)	0.22476
GraphMVP-C	0.276 (0.004)	0.168 (0.001)	0.22231

A.7.6. Case Studies

Shape Analysis (3D Diameter Prediction). Diameter is an important measure in molecule [161, 167], and genome [60] modelling. Usually, the longer the 2D diameter (longest adjacency path) is, the larger the 3D diameter (largest atomic pairwise l2 distance). However, this is not always true. Therefore, we are particularly interested in using the 2D graph to predict the 3D diameter when the 2D and 3D molecular landscapes are with large differences (as in Figure 3 and Figure 19). We formulate it as a n -class recognition problem, where n is the number of class after removing the consecutive intervals. We provide numerical results in Table 27 and more visualisation examples in Figure 20.

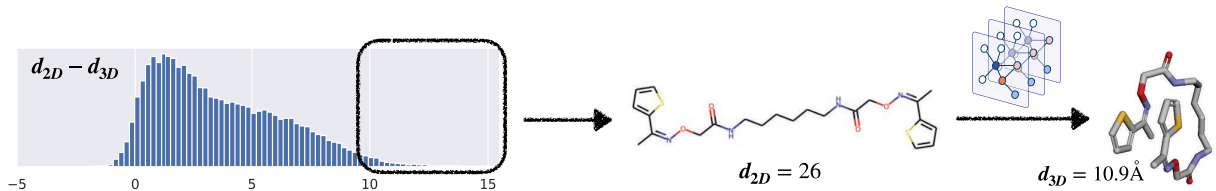


Figure 19. Molecules selection, we select the molecules that lies in the black dash box.

Table 27. Accuracy on Recognizing Molecular Spatial Diameters

Random	AttrMask	ContextPred	GPT-GNN	GraphCL	JOAOv2	MVP	MVP-G	MVP-C
38.9 (0.8)	37.6 (0.6)	41.2 (0.7)	39.2 (1.1)	38.7 (2.0)	41.3 (1.2)	42.3 (1.9)	41.9 (0.7)	42.3 (1.3)

Long-Range Donor-Acceptor Detection. Donor-Acceptor structures such as hydrogen bonds have key impacts on the molecular geometrical structures (collinear and coplanarity), and physical properties (melting point, water affinity, viscosity etc.). Usually, atom pairs such as “O...H” that are closed in the Euclidean space are considered as the donor-acceptor structures [118]. On this basis, we are particularly interested in using the 2D graph to recognize (i.e., binary classification) donor-acceptor structures which have larger ranges in the 2D adjacency (as shown in Figure 3). Similarly, we select the molecules whose donor-acceptor are close in 3D Euclidean distance but far in the 2D adjacency. We provide numerical results in Table 28. Both tables show that MVP is the MVP :)

Table 28. Accuracy on Recognizing Long-Range Donor-Acceptor Structures

Random	AttrMask	ContextPred	GPT-GNN	GraphCL	JOAOv2	MVP	MVP-G	MVP-C
77.9 (1.1)	78.6 (0.3)	80.0 (0.5)	77.5 (0.9)	79.9 (0.7)	79.2 (1.0)	80.0 (0.4)	81.5 (0.4)	80.7 (0.2)

Chirality. We have also explored other tasks such as predicting the molecular chirality, it is a challenging setting if only 2D molecular graphs are provided [186]. We found that GraphMVP brings negligible improvements due to the model capacity of SchNet. We save this in the ongoing work.



Figure 20. Molecule examples where GraphMVP successfully recognizes the 3D diameters while random initialisation fails, legends are in a format of “molecule id”-“2d diameter”-“3d diameter”.

Appendix B

Appendix for GeoSSL: Molecular Geometry Pretraining with SE(3)-Invariant Denoising Distance Matching

B.1. Benchmarks and Related Work

B.1.1. Geometric Neural Networks

Recently, geometric neural networks have been actively proposed, including SchNet [207], TFN [61], DimeNet++ [126], SE(3)-Trans [61], EGNN [202], SEGNN [22], SphereNet [159], SpinConv [214], PaiNN [208], and GemNet [125]. We reproduce most of them on the QM9 dataset as shown in Table 29. Among this, we would like to highlight two models: SchNet and PaiNN.

SchNet [206] is composed of the following key steps:

$$z_i^{(0)} = \text{embedding}(x_i), \quad z_i^{(t+1)} = \text{MLP}\left(\sum_{j=1}^n f(x_j^{(t-1)}, r_i, r_j)\right), \quad h_i = \text{MLP}(z_i^{(K)}), \quad (\text{B.1.1})$$

where K is the number of hidden layers, and

$$f(x_j, r_i, r_j) = x_j \cdot e_k(r_i - r_j) = x_j \cdot \exp(-\gamma \|\|r_i - r_j\|_2 - \mu\|_2^2) \quad (\text{B.1.2})$$

is the continuous-filter convolution layer, enabling the modeling of continuous coordinates of atoms.

PaiNN [208] is an improved work of SchNet [206]. It addresses the limitation of rotational equivariance in SchNet by embracing rotational invariance, attaining a more expressive SE(3)-equivariant neural network model.

B.1.2. Benchmark on QM9

Current work is using different optimization strategies and different data split (in terms of the splitting size). Originally there are 133,885 molecules in QM9, where 3,054 are filtered out, leading to 130,831 molecules. During the benchmark, we find that:

- The performance on QM9 is very robust to either using (1) 110K for training, 10K for val, 10,831 for test or using (2) 100K for training, 13,083 for val and 17,748 for test.
- The optimization, especially the learning rate scheduler is very critical. During the benchmarking, we find that using cosine annealing learning rate schedule [163] is generally the most robust.

For more detailed discussion on QM9, please refer to Appendix B.4. We show the benchmark results on QM9 in Table 29.

Table 29. Benchmark results on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for test. The evaluation is mean absolute error (MAE).

	Alpha ↓	Gap ↓	HOMO ↓	LUMO ↓	Mu ↓	Cv ↓	G298 ↓	H298 ↓	R2 ↓	U298 ↓	U0 ↓	Zpve ↓
SchNet	0.070	50.38	31.81	25.76	0.029	0.031	14.60	14.24	0.131	13.99	14.12	1.686
SE(3)-Trans	0.136	58.27	35.95	35.41	0.052	0.068	68.50	70.22	1.828	70.14	72.28	5.302
EGNN	0.067	48.77	28.98	24.44	0.032	0.031	11.02	11.07	0.078	10.83	10.70	1.578
DimeNet++	0.046	38.14	21.23	17.57	0.029	0.022	7.98	7.19	0.306	6.86	6.93	1.204
SphereNet	0.050	39.54	21.88	18.66	0.026	0.025	8.65	7.43	0.262	8.28	8.01	1.390
SEGNN	0.057	41.08	22.46	21.46	0.025	0.028	13.07	13.94	0.472	14.64	13.89	1.662
PaiNN	0.048	44.50	26.00	21.11	0.016	0.025	8.31	7.67	0.132	7.77	7.89	1.322

B.1.3. Related Work

We acknowledge that there is a parallel work called Protein Tertiary SSL (PTSSL) [77] working on the geometric self-supervised learning. Yet, there are some fundamental differences between theirs and ours, as listed below: **(1) Key notion on pseudo-force.** PTSSL directly applies the denoised score matching method into protein tertiary structures, yet our focus is on how the notion of pseudo-force can come into the play, which possess better generalization ability. **(2) Task setting.** PTSSL works on protein and utilize both the 2D and 3D information, and our work is purely working on the 3D geometric information. **(3) Technical novelty.** PTSSL designs the DSM objective for SSL, and what we propose is a systematic tool: using energy-based model and score matching to solve the geometric SSL problem opens a new venue in this field. **(4) Objective.** PTSSL directly designs one objective function, which is denoising from one view to the other. Ours starts from the lower bound of MI, which is symmetric in terms of the denoising directions. We believe that such symmetry are treating the two views equally, and can better reveal the mutual concept,

making the pre-trained representation more robust to the position augmentations. **(5) Empirical baseline.** PTSSL lacks the comparisons with other pre-training methods, while we compare with 7 SOTA pre-training methods, especially those driven by maximizing the MI with the same augmentations. Without such comparisons, it is hard to tell the effectiveness of the pseudo-force matching for geometric data. **(6) Score network.** Last but not least, the score network designed in PTSSL does not satisfy the SE(3) equivariant property.

B.2. An Example On The Importance of Atom Coordinates

First, it has been widely acknowledged [53] that the atom positions or molecule shapes are important factors to the quantum properties. Here we carry out an evidence example to empirically verify this. The goal here is to make predictions on 12 quantum properties in QM9.

The molecule geometric data includes two main components as input features: the atom types and atom coordinates. Other key information can be inferred accordingly, including the pairwise distances and torsion angles. We consider corruption in each of the components to empirically test their importance accordingly.

- Atom type corruption. There are in total 118 types of atom types, and the standard embedding option is to apply the one-hot encoding. In the corruption case, we replace all the atom types with a hold-out index, *i.e.*, index 119.
- Atom coordinate corruption. Originally QM9 includes atom coordinates that are in the stable state, and now we replace them with the coordinates generated with MMFF [81] from RDKit [129].

Table 30. An evidence example of molecular data. The goal is to predict 12 quantum properties (regression tasks) of 3D molecules (with 3D coordinates on each atom). The evaluation metric is MAE.

Model	Mode	Alpha ↓	Gap ↓	HOMO ↓	LUMO ↓	Mu ↓	Cv ↓	G298 ↓	H298 ↓	R2 ↓	U298 ↓	U0 ↓	Zpve ↓
SchNet	Stable Geometry	0.070	50.59	32.53	26.33	0.029	0.032	14.68	14.85	0.122	14.70	14.44	1.698
	Type Corruption	0.074	52.07	33.64	26.75	0.032	0.032	21.68	22.93	0.231	23.01	22.99	1.677
	Coordinate Corruption	0.265	110.59	79.92	78.59	0.422	0.113	57.07	58.92	18.649	60.71	59.32	5.151
PaiNN	Stable Geometry	0.048	44.50	26.00	21.11	0.016	0.025	8.31	7.67	0.132	7.77	7.89	1.322
	Type Corruption	0.057	45.61	27.22	22.16	0.016	0.025	11.48	11.60	0.181	11.15	10.89	1.339
	Coordinate Corruption	0.223	108.31	73.43	72.35	0.391	0.095	48.40	51.82	16.828	51.43	48.95	4.395

We take SchNet and PaiNN as the backbone 3D GNN models, and the results are in Table 30. We can observe that (1) Both corruption examples lead to performance decrease. (2) The atom coordinate corruption may lead to more severe performance decrease than the atom type corruption. To put this into another way is that, when we corrupt the atom types with the same hold-out type, it is equivalently to removing the atom type information. Thus, this can be viewed as using the equilibrium atom coordinates alone, and the

property prediction is comparatively robust. This observation can also be supported from the domain perspective. According to the valence bond theory, the atom type information can be implicitly and roughly inferred from the atom coordinates.

Therefore, by combining all the above observations and analysis, one can draw the conclusion that, *for molecule geometry data, the atom coordinates reveal more fundamental information for representation learning.*

B.3. Mutual Information Maximization with Energy-Based Model

In this section, we will give a detailed discussion on mutual information (MI) maximization with the energy-based model (EBM).

First, we can get a lower bound of MI. Assuming that there exist (possibly negative) constants a and b such that $a \leq H(X)$ and $b \leq H(Y)$, *i.e.*, the lower bounds to the (differential) entropies, then we have:

$$\begin{aligned} I(X; Y) &= \frac{1}{2}(H(X) + H(Y) - H(Y|X) - H(X|Y)) \\ &\geq \frac{1}{2}(a + b - H(Y|X) - H(X|Y)) \\ &\geq \frac{1}{2}(a + b) + \mathcal{L}_{\text{MI}}, \end{aligned} \tag{B.3.1}$$

where the loss \mathcal{L}_{MI} is defined as:

$$\mathcal{L}_{\text{MI}} = \frac{1}{2} \mathbb{E}_{p(\mathbf{x}, \mathbf{y})} \left[\log p(\mathbf{x}|\mathbf{y}) + \log p(\mathbf{y}|\mathbf{x}) \right]. \tag{B.3.2}$$

Empirically, we use energy-based models to model the distributions. The existence of a and b can be understood as the requirements that the two distributions ($p_{\mathbf{x}}, p_{\mathbf{y}}$) are not collapsed. Notice that to keep consistent with the notations in Section 3, we will be using \mathbf{g}_1 and \mathbf{g}_2 as the two variables. Then the goal is equivalent to optimizing the following equation:

$$\mathcal{L}_{\text{GeoSSL}} \triangleq \frac{1}{2} \mathbb{E}_{p(\mathbf{g}_1, \mathbf{g}_2)} \left[\log p(\mathbf{g}_1|\mathbf{g}_2) + \log p(\mathbf{g}_2|\mathbf{g}_1) \right]. \tag{B.3.3}$$

Thus, we transform the MI maximization problem into maximizing the summation of two conditional log-likelihoods. Such an objective function opens a wider venue for estimating MI, *e.g.*, using the EBM to estimate Equation (B.3.3).

Adaptation to Geometric Data. The 3D geometric information or the atomic coordinates are critical to molecular properties. Then based on this, we propose a geometry perturbation, which adds small noises to the atom coordinates. This geometry perturbation possesses certain motivations from both domain and machine learning perspectives. (1) From the practical experiment perspective, the statistical and systematic errors [33] on conformation estimation are unavoidable. Coordinate perturbation is a natural way to enable learning representations robust to such noises. (2) From the domain aspect, molecules are not static but in continuous motion in the 3D Euclidean space, and we can obtain a potential energy surface accordingly. We are interested in modeling the conformer, *i.e.*, the 3D coordinates with the lowest energy. However, even the conformer at the lowest energy point can have vibrations, and coordinate perturbation can better capture such movement yet with the same order of magnitude on energies. (3) As will be illustrated later, our proposed method can be simplified as denoising atomic distance matching. (4) Leveraging coordinate perturbation

for model regularization has also been empirically verified its effectiveness for supervised molecule geometric representation learning [70]. Such characteristics of molecular geometry motivate us to apply coordinate perturbation. If we take each of the two views as adding noise to the coordinates from the other view, then the objective in Equation (B.3.3) essentially states that we want to conduct coordinate denoising, as shown in Figure 21. Yet, this is not a trivial task due to the complicated geometric space (*e.g.*, 3D coordinates) reconstruction.

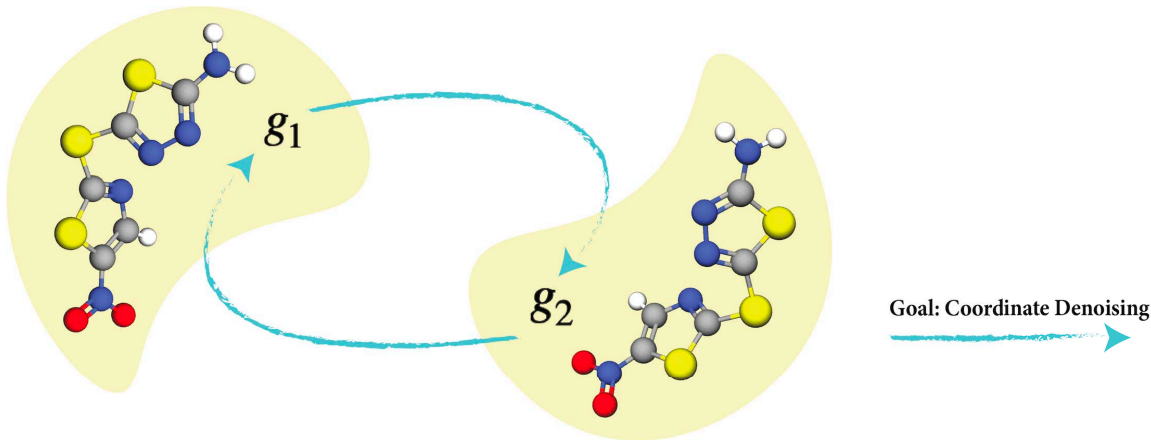


Figure 21. Pipeline for denoising coordinate matching.

B.3.1. An EBM framework for MI estimation

The lower bound in Equation (B.3.3) is composed of two conditional log-likelihood terms, and then we model the conditional likelihood with EBM. This gives us:

$$\mathcal{L}_{\text{GeoSSL-EBM}} = -\frac{1}{2} \mathbb{E}_{p(\mathbf{g}_1, \mathbf{g}_2)} \left[\log \frac{\exp(f_{\mathbf{g}_1}(\mathbf{g}_1, \mathbf{g}_2))}{A_{\mathbf{g}_1|\mathbf{g}_2}} + \log \frac{\exp(f_{\mathbf{g}_2}(\mathbf{g}_2, \mathbf{g}_1))}{A_{\mathbf{g}_2|\mathbf{g}_1}} \right], \quad (\text{B.3.4})$$

where $f_{\mathbf{g}_1}(\mathbf{g}_1, \mathbf{g}_2) = -E(\mathbf{g}_1|\mathbf{g}_2)$ and $f_{\mathbf{g}_2}(\mathbf{g}_2, \mathbf{g}_1) = -E(\mathbf{g}_2|\mathbf{g}_1)$ are the negative energy functions, and $A_{\mathbf{g}_1|\mathbf{g}_2}$ and $A_{\mathbf{g}_2|\mathbf{g}_1}$ are the corresponding partition functions. The energy functions can be flexibly defined, thus the bottleneck here is the intractable partition function due to the high cardinality. To solve this, existing methods include noise-contrastive estimation (NCE) [79] and score matching (SM) [220, 221], and we will describe how to apply them for MI maximization.

B.3.2. EBM-NCE for MI estimation

Under the EBM framework, if we solve Equation (B.3.4) with Noise-Contrastive Estimation (NCE) [79], the final objective is termed EBM-NCE, as:

$$\begin{aligned} \mathcal{L}_{\text{GeoSSL-EBM-NCE}} = & -\frac{1}{2}\mathbb{E}_{p_{\text{data}}(y)}\left[\mathbb{E}_{p_n(g_1|g_2)}[\log(1-\sigma(f_{g_1}(g_1, g_2)))] + \mathbb{E}_{p_{\text{data}}(g_1|g_2)}[\log\sigma(f_{g_1}(g_1, g_2))]\right] \\ & -\frac{1}{2}\mathbb{E}_{p_{\text{data}}(x)}\left[\mathbb{E}_{p_n(g_2|g_1)}[\log(1-\sigma(f_{g_2}(g_2, g_1)))] + \mathbb{E}_{p_{\text{data}}(g_2|g_1)}[\log\sigma(f_{g_2}(g_2, g_1))]\right]. \end{aligned} \tag{B.3.5}$$

All the detailed derivations can be found in [79]. Specifically, EBM-NCE is equivalent to the Jensen-Shannon estimation for MI, while the mathematical intuitions and derivation processes are different. Besides, it also belongs to the contrastive SSL venue. That is, it aims at aligning the positive pairs and contrasting the negative pairs.

B.3.3. EBM-SM for MI estimation: GeoSSL-DDM

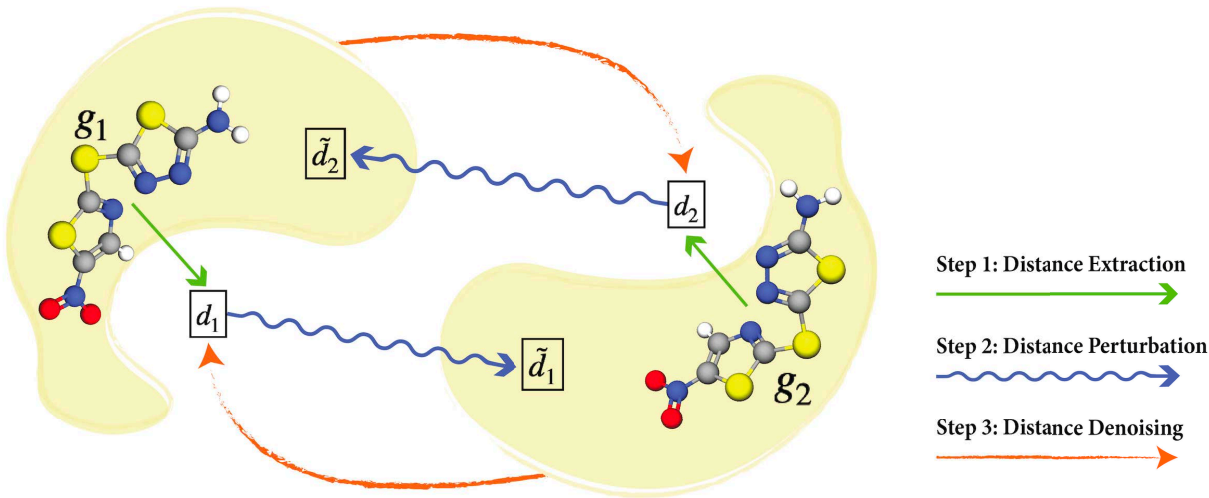


Figure 22. Pipeline for GeoSSL-DDM. The g_1 and g_2 are around the same local minima, yet with coordinate noises perturbation. Originally we want to do coordinate denoising between these two views. Then as proposed in GeoSSL-DDM, we transform it to an equivalent problem, *i.e.*, distance denoising. This figure shows the three key steps: extract the distances from the two geometric views, perform distance perturbation, and denoise the perturbed distances.

In this subsection, we will focus on geometric data like molecular geometry. Recall that we have two views: g_1 and g_2 , and the goal is to maximize the lower bound of the mutual information in Equation (B.3.3). Because the two views share the same atomic features, it

can be reduced to:

$$\begin{aligned}
\mathcal{L}_{\text{GeoSSL-EBM}} &= \frac{1}{2} \mathbb{E}_{p(\mathbf{g}_1, \mathbf{g}_2)} \left[\log p(\mathbf{g}_1 | \mathbf{g}_2) \right] + \frac{1}{2} \mathbb{E}_{p(\mathbf{g}_1, \mathbf{g}_2)} \left[\log p(\mathbf{g}_2 | \mathbf{g}_1) \right] \\
&= \frac{1}{2} \mathbb{E}_{p(\mathbf{g}_1, \mathbf{g}_2)} \left[\log p(\langle X_1, R_1 \rangle | \langle X_2, R_2 \rangle) \right] + \frac{1}{2} \mathbb{E}_{p(\mathbf{g}_1, \mathbf{g}_2)} \left[\log p(\langle X_2, R_2 \rangle | \langle X_1, R_1 \rangle) \right] \\
&= \frac{1}{2} \mathbb{E}_{p(\mathbf{g}_1, \mathbf{g}_2)} \left[\log p(R_1 | \mathbf{g}_2) \right] + \frac{1}{2} \mathbb{E}_{p(\mathbf{g}_1, \mathbf{g}_2)} \left[\log p(R_2 | \mathbf{g}_1) \right] \\
&= \frac{1}{2} \mathbb{E}_{p(\mathbf{g}_1, \mathbf{g}_2)} \left[\log \frac{\exp(f(R_1, \mathbf{g}_2))}{A_{R_1 | \mathbf{g}_2}} \right] + \frac{1}{2} \mathbb{E}_{p(\mathbf{g}_2, \mathbf{g}_1)} \left[\log \frac{\exp(f(R_2, \mathbf{g}_1))}{A_{R_2 | \mathbf{g}_1}} \right],
\end{aligned} \tag{B.3.6}$$

where the $f(\cdot)$ are the negative of energy functions, and $A_{R_1 | \mathbf{g}_2}$ and $A_{R_2 | \mathbf{g}_1}$ are the intractable partition functions. The first equation in Equation (B.3.6) results from that the two views share the same atom types. This equation can be treated as denoising the atom coordinates of one view from the geometry of the other view. In the following, we will explore how to use the score matching for solving EBM, and further transform the coordinate-aware mutual information maximization to the denoising distance matching (GeoSSL-DDM) as the final objective.

Score Definition. The two terms in Equation (4.2) are in the mirroring direction. Thus in what follows, we may as well adopt a proxy task that these two directions can be calculated separately, and take one direction for illustration, *e.g.*, $\log \frac{\exp(f(R_1, \mathbf{g}_2))}{A_{R_1 | \mathbf{g}_2}}$. The score is defined as the gradient of the log-likelihood w.r.t. the data, *i.e.*, the atom coordinates in our case. Because the normalization function is a constant w.r.t. the data, it will disappear during the score calculation. To adapt it into our setting, the score is obtained as the gradient of the negative energy function w.r.t. the atom coordinates, as:

$$s(R_1, \mathbf{g}_2) = \nabla_{R_1} \log p(R_1 | \mathbf{g}_2) = \nabla_{R_1} f(R_1, \mathbf{g}_2). \tag{B.3.7}$$

If we assume that the learned optimal energy function, *i.e.*, $f(\cdot)$, possesses certain physical or chemical information, then the score in Equation (B.3.7) can be viewed as a special form of the pseudo-force. This may require more domain-specific knowledge, and we leave this for future exploration.

Score Decomposition: From Coordinates To Distances. Through back-propagation [209], the score on atom coordinates can be further decomposed into the scores attached to pairwise

distances:

$$\begin{aligned}
s(R_1, \mathbf{g}_2)_i &= \frac{\partial f(R_1, \mathbf{g}_2)}{\partial r_{1,i}} \\
&= \sum_{j \in \mathcal{N}(i)} \frac{\partial f(R_1, \mathbf{g}_2)}{\partial d_{1,ij}} \cdot \frac{\partial d_{1,ij}}{\partial r_{1,i}} \\
&= \sum_{j \in \mathcal{N}(i)} \frac{1}{d_{1,ij}} \cdot \frac{\partial f(R_1, \mathbf{g}_2)}{\partial d_{1,ij}} \cdot (r_{1,i} - r_{1,j}) \\
&= \sum_{j \in \mathcal{N}(i)} \frac{1}{d_{1,ij}} \cdot s(\mathbf{d}_1, \mathbf{g}_2)_{ij} \cdot (r_{1,i} - r_{1,j}),
\end{aligned} \tag{B.3.8}$$

where $s(\mathbf{d}_1, \mathbf{g}_2)_{ij} \triangleq \frac{\partial f(R_1, \mathbf{g}_2)}{\partial d_{1,ij}}$. Such decomposition has a nice underlying intuition from the pseudo-force perspective: the pseudo-force on each atom can be further decomposed as the summation of pseudo-forces on the pairwise distances starting from this atom. Note that here the pairwise atoms are connected in the 3D Euclidean space, not by the covalent-bonding. Denoising Distance Matching (DDM). Then we adopt the denoising score matching (DSM) [248] to our task. To be more concrete, we take the Gaussian kernel as the perturbed noise distribution on each pairwise distance, *i.e.*, $q_\sigma(\tilde{\mathbf{d}}_1 | \mathbf{g}_2) = \mathbb{E}_{p_{\text{data}}(\mathbf{d}_1 | \mathbf{g}_2)} [q_\sigma(\tilde{\mathbf{d}}_1 | \mathbf{d}_1)]$, where σ is the deviation in Gaussian perturbation. One main advantage of using the Gaussian kernel is that the following gradient of conditional log-likelihood has a closed-form formulation: $\nabla_{\tilde{\mathbf{d}}_1} \log q_\sigma(\tilde{\mathbf{d}}_1 | \mathbf{d}_1, \mathbf{g}_2) = (\mathbf{d}_1 - \tilde{\mathbf{d}}_1) / \sigma^2$, and the goal of DSM is to train a score network to match it. This trick was first introduced in [248], and has been widely utilized in the score matching applications [218, 219].

To adapt this into our setting, this is essentially saying that we want to train a “distance network”, *i.e.*, $s_\theta(\tilde{\mathbf{d}}_1 | \mathbf{g}_2)$, to match the distance perturbation, or we can say it aims at matching the pseudo-force with the pairwise distances from another aspect. By taking the Fisher divergence as the discrepancy metric and the trick mentioned above, the estimation $s_\theta(\tilde{\mathbf{d}}_1, \mathbf{g}_2) \approx \nabla_{\tilde{\mathbf{d}}_1} \log q(\tilde{\mathbf{d}}_1 | \mathbf{d}_1, \mathbf{g}_2)$ can be simplified to the following:

$$D_F(q_\sigma(\tilde{\mathbf{d}}_1 | \mathbf{g}_2) || p_\theta(\tilde{\mathbf{d}}_1 | \mathbf{g}_2)) = \frac{1}{2} \mathbb{E}_{p_{\text{data}}(\mathbf{d}_1 | \mathbf{g}_2)} \mathbb{E}_{q_\sigma(\tilde{\mathbf{d}}_1 | \mathbf{d}_1, \mathbf{g}_2)} [\|s_\theta(\tilde{\mathbf{d}}_1, \mathbf{g}_2) - \frac{\mathbf{d}_1 - \tilde{\mathbf{d}}_1}{\sigma^2} + \|^2] + C. \tag{B.3.9}$$

Final objective. We adopt the following four model training tricks from [154, 218, 219] to stabilize the score matching training process. (1) We carry out the distance denoising at L -level of noises. (2) We add a weighting coefficient $\lambda(\sigma) = \sigma^\beta$ for each noise level, where β is the annealing factor. (3) We scale the score network by a factor of $1/\sigma$. (4) We sample the exactly same atoms from the two geometry views with masking ratio r . Finally, by considering the two directions and all the above tricks, the objective function becomes the

follows:

$$\begin{aligned} \mathcal{L}_{\text{GeoSSL-DDM}} = & \frac{1}{2L} \sum_{l=1}^L \sigma_l^\beta \mathbb{E}_{p_{\text{data}}(\mathbf{d}_1|\mathbf{g}_2)} \mathbb{E}_{q(\tilde{\mathbf{d}}_1|\mathbf{d}_1,\mathbf{g}_2)} \left[\left\| \frac{s_\theta(\tilde{\mathbf{d}}_1, \mathbf{g}_2)}{\sigma_l} - \frac{\mathbf{d}_1 - \tilde{\mathbf{d}}_1}{\sigma_l^2} \right\|_2^2 \right] \\ & + \frac{1}{2L} \sum_{l=1}^L \sigma_l^\beta \mathbb{E}_{p_{\text{data}}(\mathbf{d}_2|\mathbf{g}_1)} \mathbb{E}_{q(\tilde{\mathbf{d}}_2|\mathbf{d}_2,\mathbf{g}_1)} \left[\left\| \frac{s_\theta(\tilde{\mathbf{d}}_2, \mathbf{g}_1)}{\sigma_l} - \frac{\mathbf{d}_2 - \tilde{\mathbf{d}}_2}{\sigma_l^2} \right\|_2^2 \right]. \end{aligned} \tag{B.3.10}$$

B.3.4. Discussions

Using the energy-based model (EBM) to solve MI maximization can open a novel venue, especially for high-structured data like molecular geometry. To solve EBM, existing methods include noise-contrastive estimation (NCE) [79], score matching (SM) [221], etc. To put this under the MI maximization setting, EBM-NCE is essentially a contrastive learning method, where the goal is to align the positive pairs and contrast the negative pairs simultaneously. While EBM-SM or GeoSSL-DDM, is a generative self-supervised learning (SSL) on distance denoising, and it is especially appealing in the field for geometric data representation learning.

Further interpretation of pseudo-force. Score matching can be smoothly adopted to 3D geometric setting. Because scores are defined as gradients of the energy function with respect to the atom positions, it can be thought of a form of pseudo-forces. Following this, GeoSSL-DDM, can be viewed as a pseudo-force matching, which is more natural to the molecular structures. However, further understanding of this requires more domain knowledge in understanding or designing of the energy function. This is beyond the scope of this paper, and we would like to leave it for future exploration.

Multi-view pretraining: complementary information with 2D topological graph. Recently, there have been certain works [154] proving that 3D geometric information is useful for 2D topology. Here we want to conjecture that the reverse direction is also meaningful: 2D topology can be also useful for 3D representation. This may not seem reasonable from the domain perspective, since 2D topology can be heuristically obtained from the 3D geometry, *i.e.*, all the 2D information is redundant to 3D geometry. However, from the machine learning theory perspective [18, 64], this is still helpful in reducing the sample complexity. From a higher level perspective, we want to explicitly point out that such gap between machine learning and scientific domain has been widely existed, and it would be an interesting direction for further exploration.

B.4. Experiments

In this section, we would like to discuss the experiment details of our work. The main structure is as follows:

- In Appendix B.4.1, we introduce the computation resources.
- In Appendices B.4.2 to B.4.4, we introduce the downstream datasets.
 - Notice that because the performance of QM9 and MD17 is quite stable after fixing the seed (*e.g.*, 42), we will not run cross-validation. This also follows the main literature [159, 207, 208].
 - Yet, for LBA & LEP, these two datasets are quite small and are very sensitive to data splitting, so we pick up 5 seeds (12, 22, 32, 42, and 52) and run cross-validation on them.
- In Appendix B.4.5, we list the key hyperparameters for all the pretraining baselines and GeoSSL-DDM.
- In Appendix B.4.6, we show the empirical results using SchNet as the backbone model.

B.4.1. Computational Resources

We have around 20 V100 GPU cards for computation at an internal cluster. Each job can be finished within 3-24 hours (each job takes one single GPU card).

B.4.2. Dataset: QM9

QM9 [192] is a dataset of 134K molecules consisting of 9 heavy atoms. It includes 12 tasks that are related to the quantum properties. For example, U0 and U298 are the internal energies at 0K at 0K and 298.15K respectively, and U298 and G298 are the other two energies that can be transferred from H298 respectively. The other 8 tasks are quantum mechanics related to the DFT process. We follow [207] in preprocessing the dataset (including unit transformation for each task).

Current work is using different data split (in terms of the splitting size). Originally there are 133,885 molecules in QM9, where 3,054 are filtered out, leading to 130,831 molecules. During the benchmark, we find that the performance on QM9 is very robust to either using (1) 110K for training, 10K for val, 10,831 for test or using (2) 100K for training, 13,083 for val and 17,748 for test. In this paper, we are using option (1).

B.4.3. Dataset: MD17

MD17 [32] is a dataset on molecular dynamics simulation. It includes eight tasks, corresponding to eight organic molecules, and each task includes the molecule positions along

the potential energy surface (PES), as shown in Figure 4. The goal is to predict the energy-conserving interatomic forces for each atom at each molecule position. We list some basic statistics in Table 31. We follow [159, 208] in preprocessing the dataset (including unit transformation for each task).

Table 31. Some basic statistics on MD17.

Pretraining	Aspirin ↓	Benzene ↓	Ethanol ↓	Malonaldehyde ↓	Naphthalene ↓	Salicylic ↓	Toluene ↓	Uracil ↓
Train	1K	1K	1K	1K	1K	1K	1K	1K
Validation	1K	1K	1K	1K	1K	1K	1K	1K
Test	209,762	47,863	553,092	991,237	324,250	318,231	440,790	131,770

B.4.4. Dataset: LBA & LEP

Atom3D [236] is a newly published dataset. It gathers several core tasks for 3D molecules, including binding affinity. The binding affinity prediction is to measure the strength of binding interaction between a small molecule to the target protein. Here we will model both the small molecule and large molecule (protein) with their 3D atom coordinates provided.

Table 32. Some basic statistics on LBA & LEP. For LBA, we use split-by-sequence-identity-30: we split protein-ligand complexes such that no protein in the test dataset has more than 30% sequence identity with any protein in the training dataset. For LEP, we split the complex pairs by protein target.

Pretraining	LBA	LEP
Train	3,507	304
Validation	466	110
Test	490	104
Split	split-by-identity-30	split-by-target

During the binding process, a cavity in a protein can potentially possess suitable properties for binding a small molecule (ligand), and it is termed a pocket [222]. Because of the large volume of protein, we follow [236] by only taking the binding pocket, where there are no more than 600 atoms for each molecule and protein pair. To be more concrete, we consider two binding affinity tasks. (1) The first task is ligand binding affinity (LBA). It is gathered from [252] and the task is to predict the binding affinity strength between a small molecule and a protein pocket. (2) The second task is ligand efficacy prediction (LEP). We have a molecule bounded to pockets, and the goal is to detect if the same molecule has a higher binding affinity with one pocket compared to the other one. We list some basic statistics in Table 32.

B.4.5. Hyperparameter Specification

We list all the detailed hyperparameters in this subsection. For all the methods, we use the same optimization strategy, *i.e.*, with learning rate as 5e-4 and cosine annealing

learning rate schedule [163]. The other hyperparameters for each pretraining method are listed in Table 33. For the other hyperparameters, we are using the default hyperparameters, as attached in the codes.

Table 33. Hyperparameter specifications.

Pretraining	Hyperparameter	Value
Supervised	task	{total energy}
Type Prediction	masking ratio	{0.15, 0.3}
Distance Prediction	prediction rate	{1}
Angle Prediction	prediction rate	{1e-3, 1e-4}
RR	perturbed noise μ	{0}
	perturbed noise σ	{0.3}
	masking ratio r	{0, 0.3}
InfoNCE	perturbed noise μ	{0}
	perturbed noise σ	{0.3, 1}
	masking ratio r	{0, 0.3}
EBM-NCE	perturbed noise μ	{0}
	perturbed noise σ	{0.3, 1}
	masking ratio r	{0, 0.3}
GeoSSL-DDM	perturbed noise μ	{0}
	perturbed noise σ	{0.3}
	masking ratio r	{0, 0.3}
	L	{30, 50}
	σ_1	{0.01}
	σ_L	{10}
annealing factor β	{0.05, 0.2, 2, 5, 10}	

B.4.6. SchNet as Backbone Model

We want to highlight that some backbone models (*e.g.*, DimeNet++ and SphereNet) may perform better or on par with the PaiNN, as shown in Table 29. Yet they will be out of GPU memory. Thus, considering all (including the model performance, computation efficiency, and memory cost) together, we adopt PaiNN as the backbone model in the main paper.

In this section, we carry out experiments using SchNet as the backbone model. We follow the same process as in Section 5, *i.e.*, we compare our method with one randomly-initialized and seven pretraining baselines. The results on QM9, MD17, LBA and LEP are in Tables 34 to 36 accordingly. From these three tables, we can observe that in general, GeoSSL-DDM can reach the most optimal results, yielding 21 best performance in 22 downstream tasks, and can reach comparative performance on the remaining task (within top 2 model). This can largely support the effectiveness of our proposed method, GeoSSL-DDM. In addition, we also want to mention that a lot of pretraining tasks show the negative transfer issue. Comparing to the results in Section 5, we conjecture that this is related to the task (both pretraining and downstream tasks) and the backbone model. Yet, this is beyond the scope of our work, and we would like to leave this as a future direction.

Table 34. Downstream results on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for test. The evaluation is mean absolute error, and the best results are in **bold**.

Pretraining	Alpha ↓	Gap ↓	HOMO ↓	LUMO ↓	Mu ↓	Cv ↓	G298 ↓	H298 ↓	R2 ↓	U298 ↓	U0 ↓	Zpve ↓
–	0.070	50.59	32.53	26.33	0.029	0.032	14.68	14.85	0.122	14.70	14.44	1.698
Supervised	0.070	51.34	32.62	27.61	0.030	0.032	14.08	14.09	0.141	14.13	13.25	1.727
Type Prediction	0.084	56.07	34.55	30.65	0.040	0.034	18.79	19.39	0.201	19.29	18.86	2.001
Distance Prediction	0.068	49.34	31.18	25.52	0.029	0.032	13.93	13.59	0.122	13.64	13.18	1.676
Angle Prediction	0.084	57.01	37.51	30.92	0.037	0.034	15.81	15.89	0.149	16.41	15.76	1.850
3D InfoGraph	0.076	53.33	33.92	28.55	0.030	0.032	15.97	16.28	0.117	16.17	15.96	1.666
GeoSSL-RR	0.073	52.57	34.44	28.41	0.033	0.038	15.74	16.11	0.194	15.58	14.76	1.804
GeoSSL-InfoNCE	0.075	53.00	34.29	27.03	0.029	0.033	15.67	15.53	0.125	15.79	14.94	1.675
GeoSSL-EBM-NCE	0.073	52.86	33.74	28.07	0.031	0.032	14.02	13.65	0.121	13.70	13.45	1.677
GeoSSL-DDM (ours)	0.066	48.59	30.83	25.27	0.028	0.031	13.06	12.33	0.117	12.48	12.06	1.631

Table 35. Downstream results on 8 force prediction tasks from MD17. We take 1K for training, 1K for validation, and the number of molecules for test are varied among different tasks, ranging from 48K to 991K. The evaluation is mean absolute error, and the best results are in **bold**.

Pretraining	Aspirin ↓	Benzene ↓	Ethanol ↓	Malonaldehyde ↓	Naphthalene ↓	Salicylic ↓	Toluene ↓	Uracil ↓
–	1.196	0.404	0.542	0.879	0.534	0.786	0.562	0.730
Supervised	1.863	0.413	0.512	1.254	0.846	1.005	0.529	0.899
Type Prediction	1.293	0.787	0.547	0.879	1.030	1.076	0.614	0.738
Distance Prediction	1.414	0.453	0.845	1.371	0.591	0.819	0.588	0.993
Angle Prediction	3.030	0.450	0.485	0.845	1.112	1.214	0.791	1.016
3D InfoGraph	1.545	0.448	0.640	1.080	0.827	1.096	0.735	0.760
GeoSSL-RR	1.878	0.450	0.690	2.255	0.960	1.382	0.784	1.188
GeoSSL-InfoNCE	1.286	0.396	0.512	1.007	0.778	1.060	0.667	0.933
GeoSSL-EBM-NCE	1.271	0.400	0.570	0.972	0.605	0.862	0.576	0.790
GeoSSL-DDM (ours)	1.176	0.368	0.434	0.779	0.460	0.700	0.561	0.679

Table 36. Downstream results on 2 binding affinity tasks. We select three evaluation metrics for LBA: the root mean squared error (RMSD), the Pearson correlation (R_p) and the Spearman correlation (R_S). LEP is a binary classification task, and we use the area under the curve for receiver operating characteristics (ROC) and precision-recall (PR) for evaluation. We run cross-validation with 5 seeds, and the best results are in **bold**.

Pretraining	LBA			LEP	
	RMSD ↓	R_P ↑	R_C ↑	ROC ↑	PR ↑
–	1.489 ± 0.02	0.522 ± 0.01	0.501 ± 0.01	0.436 ± 0.03	0.369 ± 0.02
Supervised	1.477 ± 0.04	0.528 ± 0.02	0.503 ± 0.03	0.462 ± 0.05	0.392 ± 0.03
Type Prediction	1.483 ± 0.04	0.498 ± 0.03	0.481 ± 0.03	0.570 ± 0.04	0.509 ± 0.07
Distance Prediction	1.461 ± 0.06	0.535 ± 0.04	0.512 ± 0.04	0.502 ± 0.06	0.415 ± 0.05
Angle Prediction	1.499 ± 0.01	0.475 ± 0.01	0.462 ± 0.02	0.532 ± 0.06	0.449 ± 0.03
3D InfoGraph	1.467 ± 0.06	0.526 ± 0.03	0.500 ± 0.03	0.515 ± 0.05	0.412 ± 0.04
GeoSSL-RR	–	–	–	0.439 ± 0.04	0.365 ± 0.02
GeoSSL-InfoNCE	1.528 ± 0.05	0.483 ± 0.02	0.464 ± 0.02	0.588 ± 0.06	0.523 ± 0.05
GeoSSL-EBM-NCE	1.499 ± 0.03	0.509 ± 0.02	0.498 ± 0.02	0.493 ± 0.07	0.429 ± 0.06
GeoSSL-DDM (ours)	1.432 ± 0.02	0.550 ± 0.02	0.529 ± 0.02	0.633 ± 0.03	0.541 ± 0.03

B.5. Ablation Studies

B.5.1. The Effect of Annealing Factor in GeoSSL-DDM

Among all the hyperparameters (see Table 33) for GeoSSL-DDM, we find that the annealing factor is one of the most sensitive ones. Annealing factor β is applied on the weighting coefficient $\lambda(\sigma) = \sigma^\beta$. In this section, we carry out an ablation study to verify this by pretraining GeoSSL-DDM with annealing factors at five different scales.

Table 37. Ablation study on the effect of annealing factor β on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for test. The backbone model is PaiNN, and the evaluation is the mean absolute error.

β	Alpha ↓	Gap ↓	HOMO ↓	LUMO ↓	Mu ↓	Cv ↓	G298 ↓	H298 ↓	r2 ↓	U298 ↓	U0 ↓	Zpve ↓
0.05	0.047	40.10	23.71	19.40	0.016	0.025	7.72	7.15	0.131	7.30	7.07	1.312
0.2	0.046	40.22	23.48	19.42	0.015	0.024	7.65	7.09	0.122	6.99	6.92	1.307
2	0.049	40.88	23.96	19.89	0.015	0.029	8.60	7.95	0.136	7.81	7.62	1.357
5	0.056	45.01	26.36	20.68	0.016	0.030	9.97	9.56	0.136	9.81	9.46	1.597
10	0.055	44.41	26.87	21.13	0.015	0.027	10.42	9.48	0.133	9.42	9.47	1.592

As can be observed in Table 37, the models are more stable with smaller annealing values (*e.g.*, 0.2 and 0.05). With large annealing values, the model performance can degrade drastically.

B.5.2. The Effect on the Number of Noise Layers in GeoSSL-DDM

Another important hyperparameter listed in Table 33 is the number of noise layers, L . Here we conduct an ablation study on it, and the results are shown in Table 38.

In Table 38, we can observe that in general, GeoSSL-DDM can attain better performance with more denoising layers. This is in fact consistent with that in vision applications [221]. Promisingly, even with smaller L (*e.g.*, $L = 1$), GeoSSL-DDM can still achieve a modest improvement to some extent.

Table 38. Ablation study on the effect of the noise layer L on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for test. The backbone model is PaiNN, and the evaluation is the mean absolute error.

L	Alpha ↓	Gap ↓	HOMO↓	LUMO ↓	Mu ↓	Cv ↓	G298 ↓	H298 ↓	r2 ↓	U298 ↓	U0 ↓	Zpve ↓
– (random init)	0.048	44.50	26.00	21.11	0.016	0.025	8.31	7.67	0.132	7.77	7.89	1.322
1	0.052	42.75	25.12	20.46	0.015	0.027	9.40	9.08	0.121	8.73	8.80	1.585
30	0.048	40.08	23.95	19.71	0.016	0.025	8.16	7.48	0.137	7.42	7.17	1.311
50	0.046	40.22	23.48	19.42	0.015	0.024	7.65	7.09	0.122	6.99	6.92	1.307

B.6. Strong Model Robustness with Random Seeds

To further illustrate that our proposed GeoSSL-DDM is robust and insensitive to certain random seeds, we further provide the downstream results with more random seeds. We list the key details as follows:

- **Dataset.** We conduct downstream experiments with random seeds on two datasets: QM9 and MD17.
- **Backbone models.** We run two backbone models: PaiNN in Appendix B.6.1 and SchNet in Appendix B.6.2.
- **Seeds.** Up till now, for both the main tables (Tables 6, 7, 34 and 35) and ablation studies (in Appendix B.5), we use a fixed seed 42. In this section, we provide results with two additional seeds 22 and 32.
- **Baselines.** We here compare against the most optimal baselines: random initialization (without any pretraining), distance prediction, representation reconstruction (RR), and EBM-NCE.
- **Reported results.** We report both the mean and standard deviation with seeds 22, 32, and 42 for all the experiments.

B.6.1. PaiNN

Here we take the PaiNN as the backbone model. The results on QM9 and MD17 are reported in Tables 39 and 40 respectively. Such empirical results match with the main result in Tables 6 and 7, and they do verify that our proposed GeoSSL-DDM is indeed learning a more robust representation.

Table 39. Downstream results on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for test. The evaluation metric is mean absolute error, and the best results are in **bold**. We report both the mean and standard deviation for seeds 22, 32, and 42.

Pretraining	Alpha ↓	Gap ↓	HOMO ↓	LUMO ↓	Mu ↓	Cv ↓	G298 ↓	H298 ↓	R2 ↓	U298 ↓	U0 ↓	Zpve ↓
-	0.050 ± 0.00	44.41 ± 0.75	25.81 ± 0.17	21.50 ± 0.31	0.016 ± 0.00	0.025 ± 0.00	8.27 ± 0.17	7.78 ± 0.24	0.134 ± 0.01	7.82 ± 0.04	7.93 ± 0.23	1.310 ± 0.01
Supervised	0.049 ± 0.00	44.27 ± 0.78	26.90 ± 0.25	21.85 ± 0.09	0.017 ± 0.00	0.026 ± 0.00	8.94 ± 0.11	8.54 ± 0.11	0.167 ± 0.01	8.40 ± 0.13	8.25 ± 0.07	1.381 ± 0.05
Distance Prediction	0.062 ± 0.00	51.96 ± 4.53	28.38 ± 0.80	22.63 ± 0.23	0.234 ± 0.30	0.070 ± 0.05	12.39 ± 0.27	12.63 ± 0.23	0.308 ± 0.23	12.28 ± 0.45	12.08 ± 0.20	1.745 ± 0.07
GeoSSL-RR	0.047 ± 0.00	44.70 ± 0.69	25.50 ± 0.06	21.35 ± 0.41	0.015 ± 0.00	0.025 ± 0.00	8.57 ± 0.23	8.03 ± 0.26	0.141 ± 0.01	8.21 ± 0.93	7.75 ± 0.11	1.317 ± 0.03
GeoSSL-InfoNCE	0.055 ± 0.00	45.37 ± 0.20	26.83 ± 0.10	21.95 ± 0.24	0.017 ± 0.00	0.044 ± 0.03	17.22 ± 11.44	17.97 ± 12.47	0.514 ± 0.55	17.79 ± 12.86	17.42 ± 12.59	1.902 ± 0.58
GeoSSL-EBM-NCE	0.049 ± 0.00	44.18 ± 0.31	26.15 ± 0.17	21.77 ± 0.23	0.015 ± 0.00	0.026 ± 0.00	8.79 ± 0.20	8.25 ± 0.14	0.131 ± 0.00	8.21 ± 0.15	8.27 ± 0.26	1.428 ± 0.02
GeoSSL-DDM (ours)	0.045 ± 0.00	40.29 ± 0.29	23.42 ± 0.09	19.52 ± 0.13	0.015 ± 0.00	0.025 ± 0.00	7.75 ± 0.16	7.17 ± 0.13	0.124 ± 0.00	7.15 ± 0.15	6.98 ± 0.11	1.292 ± 0.01

Table 40. Downstream results on 8 force prediction tasks from MD17. We take 1K for training, 1K for validation, and the number of molecules for test are varied among different tasks, ranging from 48K to 991K. The evaluation is mean absolute error, and the best results are in **bold**. We report both the mean and standard deviation for seeds 22, 32, and 42.

Pretraining	Aspirin ↓	Benzene ↓	Ethanol ↓	Malonaldehyde ↓	Naphthalene ↓	Salicylic ↓	Toluene ↓	Uracil ↓
-	0.559 ± 0.01	0.052 ± 0.00	0.220 ± 0.01	0.338 ± 0.00	0.138 ± 0.00	0.293 ± 0.00	0.156 ± 0.00	0.201 ± 0.01
Supervised	0.507 ± 0.02	0.180 ± 0.08	0.312 ± 0.03	0.480 ± 0.04	0.299 ± 0.11	0.469 ± 0.07	0.238 ± 0.03	0.435 ± 0.02
Distance Prediction	1.701 ± 0.20	0.146 ± 0.04	0.368 ± 0.07	0.757 ± 0.23	0.734 ± 0.07	1.493 ± 0.35	0.340 ± 0.04	0.766 ± 0.29
GeoSSL-RR	0.527 ± 0.04	0.053 ± 0.00	0.223 ± 0.01	0.342 ± 0.02	0.136 ± 0.01	0.296 ± 0.01	0.149 ± 0.01	0.190 ± 0.01
GeoSSL-InfoNCE	0.999 ± 0.11	0.108 ± 0.03	0.263 ± 0.02	0.469 ± 0.06	0.415 ± 0.16	0.516 ± 0.08	0.189 ± 0.01	0.506 ± 0.04
GeoSSL-EBM-NCE	0.724 ± 0.10	0.105 ± 0.02	0.267 ± 0.02	0.479 ± 0.03	0.362 ± 0.13	0.517 ± 0.13	0.241 ± 0.07	0.468 ± 0.02
GeoSSL-DDM (ours)	0.439 ± 0.01	0.051 ± 0.00	0.170 ± 0.00	0.290 ± 0.01	0.133 ± 0.01	0.267 ± 0.00	0.122 ± 0.00	0.192 ± 0.01

B.6.2. SchNet

Here we take the SchNet as the backbone model. The results on QM9 and MD17 are reported in Tables 41 and 42 respectively. Such empirical results match with the main result in Tables 34 and 35, and they do verify that our proposed GeoSSL-DDM is indeed learning a more robust representation.

Table 41. Downstream results on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for test. The evaluation is mean absolute error, and the best results are in **bold**. We report both the mean and standard deviation for seeds 22, 32, and 42.

Pretraining	Alpha ↓	Gap ↓	HOMO↓	LUMO ↓	Mu ↓	Cv ↓	G298 ↓	H298 ↓	R2 ↓	U298 ↓	U0 ↓	Zpve ↓
–	0.070 ± 0.00	50.19 ± 0.54	32.35 ± 0.35	26.11 ± 0.31	0.029 ± 0.00	0.032 ± 0.00	14.66 ± 0.12	14.67 ± 0.25	0.129 ± 0.01	14.40 ± 0.21	14.14 ± 0.22	1.699 ± 0.02
Supervised	0.069 ± 0.00	51.07 ± 0.34	32.20 ± 0.37	27.42 ± 0.17	0.030 ± 0.00	0.032 ± 0.00	14.08 ± 0.11	13.92 ± 0.18	0.142 ± 0.00	13.96 ± 0.14	13.41 ± 0.12	1.715 ± 0.03
Distance Prediction	0.067 ± 0.00	49.59 ± 0.32	31.17 ± 0.04	26.08 ± 0.40	0.029 ± 0.00	0.032 ± 0.00	13.81 ± 0.10	13.45 ± 0.11	0.129 ± 0.01	13.49 ± 0.18	13.10 ± 0.13	1.678 ± 0.02
GeoSSL-RR	0.078 ± 0.00	53.36 ± 0.56	34.83 ± 0.47	29.84 ± 1.43	0.034 ± 0.00	0.036 ± 0.00	16.84 ± 0.90	15.32 ± 0.67	0.203 ± 0.01	16.43 ± 0.92	15.68 ± 0.72	1.809 ± 0.01
GeoSSL-InfoNCE	0.075 ± 0.00	53.27 ± 0.20	33.80 ± 0.40	27.64 ± 0.47	0.029 ± 0.00	0.033 ± 0.00	15.59 ± 0.06	15.40 ± 0.09	0.125 ± 0.00	15.34 ± 0.32	15.24 ± 0.22	1.670 ± 0.01
GeoSSL-EBM-NCE	0.072 ± 0.00	52.64 ± 0.37	33.47 ± 0.24	28.01 ± 0.41	0.031 ± 0.00	0.032 ± 0.00	13.67 ± 0.25	13.58 ± 0.10	0.124 ± 0.00	13.52 ± 0.14	13.42 ± 0.12	1.661 ± 0.01
GeoSSL-DDM (ours)	0.066 ± 0.00	48.78 ± 0.15	30.38 ± 0.32	25.52 ± 0.23	0.028 ± 0.00	0.031 ± 0.00	12.80 ± 0.19	12.36 ± 0.09	0.113 ± 0.00	12.53 ± 0.04	12.12 ± 0.06	1.637 ± 0.01

Table 42. Downstream results on 8 force prediction tasks from MD17. We take 1K for training, 1K for validation, and the number of molecules for test are varied among different tasks, ranging from 48K to 991K. The evaluation metric is mean absolute error, and the best results are in **bold**. We report the both the mean and standard deviation for seeds 22, 32, and 42.

Pretraining	Aspirin ↓	Benzene ↓	Ethanol ↓	Malonaldehyde ↓	Naphthalene ↓	Salicylic ↓	Toluene ↓	Uracil ↓
–	1.418 ± 0.28	0.406 ± 0.00	0.528 ± 0.01	0.908 ± 0.04	0.613 ± 0.06	0.854 ± 0.08	0.575 ± 0.01	0.717 ± 0.02
Supervised	1.714 ± 0.21	0.423 ± 0.04	0.517 ± 0.01	1.127 ± 0.16	0.713 ± 0.14	1.114 ± 0.08	0.578 ± 0.08	0.832 ± 0.12
Distance Prediction	1.756 ± 0.24	0.483 ± 0.02	0.813 ± 0.02	1.458 ± 0.08	0.795 ± 0.15	1.074 ± 0.19	0.691 ± 0.08	1.116 ± 0.09
GeoSSL-RR	2.082 ± 0.20	0.563 ± 0.09	0.740 ± 0.05	1.795 ± 0.35	0.910 ± 0.07	1.525 ± 0.24	0.847 ± 0.14	1.159 ± 0.08
GeoSSL-InfoNCE	1.375 ± 0.07	0.432 ± 0.03	0.560 ± 0.05	1.101 ± 0.12	0.797 ± 0.02	1.029 ± 0.02	0.706 ± 0.03	0.934 ± 0.02
GeoSSL-EBM-NCE	1.297 ± 0.03	0.404 ± 0.00	0.569 ± 0.00	1.005 ± 0.04	0.580 ± 0.02	0.840 ± 0.02	0.581 ± 0.02	0.839 ± 0.04
GeoSSL-DDM (ours)	1.333 ± 0.23	0.379 ± 0.01	0.466 ± 0.04	0.732 ± 0.03	0.566 ± 0.13	0.824 ± 0.16	0.566 ± 0.05	0.682 ± 0.07

B.7. Comparison with a Parallel Work

We note that there is a parallel work introduced in [278], which also explores the effect of denoising for geometric data pretraining. That work is different from GeoSSL-DDM and we here summarize the main differences as follows:

- The parallel work as presented in [278] is similar to that of denoising score matching (DSM) as introduced in [248], *i.e.*, with only one layer of denoising in score matching. On the contrary, our model has multiple denoising layers, which is much closer to the NCSN [218], where the number of noise layers has been proven to be important to the effectiveness of the denoising score matching models. We here also empirically verify the above analysis. That is, we present the experimental results in Table 38, where $L = 1$ is equivalent to the method in [278]. We can observe that with layer number $L = 1$ (namely the third row of the table), the performance does increase in some cases, which matches with the observation in [278]. Nevertheless, the results in Table 38 clearly indicate that with larger L , the model can attain further error reduction and improve model robustness.
- Theoretically, the work in [278] specifically aims at the application task of representation learning in geometric pretraining, through a straightforward adaption of denoising score matching from vision. In contrast, our GeoSSL-DDM approach indeed provides a very general framework that leverages energy-based model (EBM) for mutual information (MI) maximization for geometric data pretraining. As such, GeoSSL-DDM can be easily replaced by other EBM models such as the GFlowNet network [14] to better capture the multi-mode distributions in geometric data during pretraining (please see Section 6 for more discussion).

Appendix C

Appendix for MoleculeSDE: A Group Symmetric Stochastic Differential Equation Model for Molecule Multi-modal Pretraining

C.1. Comparison to Related Works

In Table 43, we provide a comprehensive overview of existing works on single-modal and multi-modal pretraining methods. We categorize the pretraining methods into generative and contrastive learning methods

Table 43. Comparison between MoleculeSDE and existing graph SSL methods.

Pre-training	2D Topology		3D Conformation		2D Topology and 3D Conformation	
	Generative	Contrastive	Generative	Contrastive	Generative	Contrastive
AttrMask [99, 140]	✓	-	-	-	-	-
InfoGraph [227, 247]	-	✓	-	-	-	-
ContexPred [99]	-	✓	-	-	-	-
GraphCL [274]	-	✓	-	-	-	-
Atom Type Prediction [145]	-	-	✓	-	-	-
Distance Prediction [58, 145]	-	-	✓	-	-	-
Angle Prediction [58, 145]	-	-	✓	-	-	-
3D Infograph [145]	-	-	-	✓	-	-
MI-RR Prediction [145]	-	-	✓	-	-	-
MI-InfoNCE Prediction [145]	-	-	-	✓	-	-
MI-EBM-NCE Prediction [145]	-	-	-	✓	-	-
GeoSSL-1L [278]	-	-	✓	-	-	-
GeoSSL [145]	-	-	✓	-	-	-
3D InfoMax [223]	-	-	-	-	-	✓
GraphMVP [153]	-	-	-	-	✓	✓
GraphMVP-C [153]	-	✓	-	-	✓	✓
GraphMVP-G [153]	✓	-	-	-	✓	✓
MoleculeSDE (ours)	-	-	-	-	✓	✓

C.2. Group Symmetry and Local Frame

C.2.1. SE(3)/E(3) Group action and representations

In this article, a 3D molecular graph is represented by a 3D point cloud. The corresponding symmetry group is $SE(3)$, which consists of translations and rotations. Recall that we define the notion of equivariance functions in \mathbf{R}^3 in the main text through group actions. Formally, the group $SE(3)$ is said to act on \mathbf{R}^3 if there is a mapping $\phi : SE(3) \times \mathbf{R}^3 \rightarrow \mathbf{R}^3$ satisfying the following two conditions:

(1) if $e \in SE(3)$ is the identity element, then

$$\phi(e, \mathbf{r}) = \mathbf{r} \quad \text{for } \forall \mathbf{r} \in \mathbf{R}^3.$$

(2) if $g_1, g_2 \in SE(3)$, then

$$\phi(g_1, \phi(g_2, \mathbf{r})) = \phi(g_1 g_2, \mathbf{r}) \quad \text{for } \forall \mathbf{r} \in \mathbf{R}^3.$$

Then, there is a natural $SE(3)$ action on vectors \mathbf{r} in \mathbf{R}^3 by translating \mathbf{r} and rotating \mathbf{r} for multiple times. For $g \in SE(3)$ and $\mathbf{r} \in \mathbf{R}^3$, we denote this action by $g\mathbf{r}$. Once the notion of group action is defined, we say a function $f : \mathbf{R}^3 \rightarrow \mathbf{R}^3$ that transforms $\mathbf{r} \in \mathbf{R}^3$ is equivariant if:

$$f(g\mathbf{r}) = gf(\mathbf{r}), \quad \text{for } \forall \mathbf{r} \in \mathbf{R}^3.$$

On the other hand, $f : \mathbf{R}^3 \rightarrow \mathbf{R}^1$ is invariant, if f is independent of the group actions:

$$f(g\mathbf{r}) = f(\mathbf{r}), \quad \text{for } \forall \mathbf{r} \in \mathbf{R}^3.$$

For some scenarios, our problem is chiral sensitive. That is, after mirror reflecting a 3D molecule, the properties of the molecule may change dramatically. In these cases, it's crucial to include reflection transformations into consideration. More precisely, we say an $SE(3)$ equivariant function f is **reflection anti-symmetric**, if:

$$f(\rho\mathbf{r}) \neq f(\mathbf{r}), \tag{C.2.1}$$

for some reflections $\rho \in E(3)$.

C.2.2. Equivariant Frames

Frame is a popular terminology in science areas. In physics, frame is equivalent to a coordinate system. For example, we may assign a frame to all observers, although different observers may collect different data under different frames, the underlying physics law should be the same. In other words, denote the physics law by f , then f should be an equivariant function.

Since there are three orthogonal directions in \mathbf{R}^3 , a frame in \mathbf{R}^3 consists of three orthogonal vectors:

$$F = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3).$$

Once equipped with a frame (coordinate system), we can project all geometric quantities to this frame. For example, an abstract vector $\mathbf{r} \in \mathbf{R}^3$ can be written as $\mathbf{r} = (r_1, r_2, r_3)$ under frame F , if: $\mathbf{r} = r_1\mathbf{e}_1 + r_2\mathbf{e}_2 + r_3\mathbf{e}_3$. An equivariant frame further requires the three orthonormal vectors in $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ to be equivariant. Intuitively, an equivariant frame will transform according to the global rotation or translation of the whole system. Once equipped with an equivariant frame, we can project equivariant vectors into this frame:

$$\mathbf{r} = \tilde{r}_1\mathbf{e}_1 + \tilde{r}_2\mathbf{e}_2 + \tilde{r}_3\mathbf{e}_3. \quad (\text{C.2.2})$$

We call the process of $\mathbf{r} \rightarrow \tilde{\mathbf{r}} := (\tilde{r}_1, \tilde{r}_2, \tilde{r}_3)$ the **projection** operation. Since $\tilde{r}_i = \mathbf{e}_i \cdot \mathbf{r}_i$ is expressed as an inner product between equivariant vectors, we know that $\tilde{\mathbf{r}}$ consists of scalars.

In this article, we assign an equivariant frame to each node/edge, therefore we call them the local frames. Given two atoms with 3D positions $(\mathbf{r}_i, \mathbf{r}_j)$, we can find the atom (denoted by \mathbf{r}_k) that is nearest to the center of $(\mathbf{r}_i, \mathbf{r}_j)$ by KNN algorithms. Then the equivariant frame is defined by:

$$\text{Local-Frame}(\mathbf{r}_i, \mathbf{r}_j) := \mathbf{Gram-Schmidt}\{\mathbf{r}_i - \mathbf{r}_j, \mathbf{r}_i - \mathbf{r}_k, (\mathbf{r}_i - \mathbf{r}_j) \times (\mathbf{r}_i - \mathbf{r}_k)\}. \quad (\text{C.2.3})$$

The Gram-Schmidt orthogonalization makes sure that the $\text{Local-Frame}(\mathbf{r}_i, \mathbf{r}_j)$ is orthonormal. Reflection Anti-Symmetric. Since we implement the cross product \times for building the local frames, the third vector in the frame is a pseudo-vector. Then, the **projection** operation is not invariant under reflections (the inner product between a vector and a pseudo-vector change signs under reflection). Therefore, our model is able to discriminate two 3D geometries with different chirality.

Our local frames also enable us to output equivariant vectors by multiplying scalars (v_1, v_2, v_3) with the frame: $\mathbf{v} = v_1 \cdot \mathbf{e}_1 + v_2 \cdot \mathbf{e}_2 + v_3 \cdot \mathbf{e}_3$. It's easy to check that \mathbf{v} is a $SE(3)$ equivariant (reflection anti-symmetric) vector.

C.3. Denoising Score Matching

C.3.1. Energy-Based Model (EBM)

Energy-based model (EBM) is a powerful tool for modeling the data distribution. The formulation is:

$$p_\theta(\mathbf{x}) = \frac{\exp(-E(\mathbf{x}))}{A} = \frac{\exp(f(\mathbf{x}))}{A}, \quad (\text{C.3.1})$$

where the bottleneck is the intractable partition function $A = \int_{\mathbf{x}} \exp(-E(\mathbf{x}))d\mathbf{x}$. Recently, there have been big progress [220] in solving such an intractable function, including contrastive divergence [89], score matching [104, 221], and noise contrastive estimation [79].

C.3.2. Score Matching

There exists a family of solutions called **score matching** (SM) to solve Equation (C.3.1). The core idea of SM is that for the generative task, we do not need to directly estimate the density, but we just need to know the score or gradient of the data distribution, *i.e.*, $\nabla_{\mathbf{x}} \log p(\mathbf{x})$. Then a Markov chain Monte Carlo (MCMC) strategy can be adopted for data generation.

Score. The score is defined as the gradient of log-likelihood w.r.t. the data \mathbf{x} :

$$s_{\theta}(\mathbf{x}) = \nabla_{\mathbf{x}} \log p_{\theta}(\mathbf{x}) = -\nabla_{\mathbf{x}} E(\mathbf{x}) - \nabla_{\mathbf{x}} \log A = -\nabla_{\mathbf{x}} E(\mathbf{x}) = \nabla_{\mathbf{x}} f(\mathbf{x}). \quad (\text{C.3.2})$$

Thus, by taking the score, *i.e.*, the gradient w.r.t. the data, the partition function term will disappear since it is a constant. The SM transforms the density estimation problem into a score (gradient) matching problem: if the first-order gradient of function ($s_{\theta}(x)$) can match, then the learned model distribution with EBM is able to capture the data distribution precisely.

Explicit Score Matching (ESM). For training the model, originally, SM [104] applies the Fisher divergence to measure the discrepancy between data distribution and model distribution, terms explicit score matching (ESM):

$$\begin{aligned} D_F(p_{\text{data}}(\mathbf{x})||p_{\theta}(\mathbf{x})) &= \frac{1}{2} \mathbb{E}_{p_{\text{data}}} \|\nabla_{\mathbf{x}} \log p_{\theta}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})\|_2^2 \\ &= \frac{1}{2} \mathbb{E}_{p_{\text{data}}} \|s_{\theta}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})\|_2^2. \end{aligned} \quad (\text{C.3.3})$$

The expectation w.r.t. $p_{\text{data}}(\mathbf{x})$ can be approximated using Monte Carlo sampling, yet the second term of Equation (C.3.3) is intractable to compute since it needs to know $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$. There are multiple solutions to this, including Implicit Score matching (ISM) [104] which rewrites Equation (C.3.3) using integration by parts; the other appealing solution is the denoising score matching (DSM). Both are introduced below.

Implicit Score Matching (ISM). It is still impractical to calculate Equation (C.3.3) due to the $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$ term. Under certain conditions [104], the Fisher divergence can be rewritten using integration by parts, and we can turn it to the implicit score matching (ISM), *i.e.*, ESM to ISM:

$$\begin{aligned} \mathbb{E}_{p_{\text{data}}} [\|\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - s_{\theta}(\mathbf{x})\|^2] &= \mathbb{E}_{p_{\text{data}}} \left[\frac{1}{2} (\nabla_{\mathbf{x}} E_{\theta}(\mathbf{x}))^2 + \text{tr}(\nabla_{\mathbf{x}}^2 E_{\theta}(\mathbf{x})) \right] + C \\ &= \mathbb{E}_{p_{\text{data}}} \left[\frac{1}{2} (\|s_{\theta}(\mathbf{x})\|^2 + \text{tr}(\nabla_{\mathbf{x}} s_{\theta}(\mathbf{x}))) \right] + C, \end{aligned} \quad (\text{C.3.4})$$

where C is the constant. The drawback of Equation (C.3.4) is that it requires computing the Trace of Hessian. It is computationally expensive as the computation of the full second derivatives is quadratic in the dimensionality of data.

Denoising Score Matching (DSM). Along this line, denoising score matching (DSM) [248] proposes an elegant solution by connecting the SM with denoising autoencoder. It first perturbs the data with a noise distribution, *i.e.*, $q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})$, and the goal is to use SM to approximate the $q_\sigma(\tilde{\mathbf{x}}) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})]$ with a model distribution $p_\theta(\tilde{\mathbf{x}})$. DSM [248] then calculates the Fisher divergence between the perturbed data distribution and perturbed model distribution, which leads to the following equation:

$$\begin{aligned} D_F(q_\sigma(\tilde{\mathbf{x}})||p_\theta(\tilde{\mathbf{x}})) &= \frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}})}\left[\|\nabla_{\tilde{\mathbf{x}}}\log q_\sigma(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}}\log p_\theta(\tilde{\mathbf{x}})\|^2\right] \\ &= \frac{1}{2}\mathbb{E}_{q_\sigma(\mathbf{x},\tilde{\mathbf{x}})}\left[\|\nabla_{\tilde{\mathbf{x}}}\log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) - s_\theta(\tilde{\mathbf{x}})\|^2\right] + C. \end{aligned} \quad (\text{C.3.5})$$

The detailed derivation of Equation (C.3.5) can be found in Appendix C.3.3. This is an elegant solution because under the Gaussian kernel, *i.e.*, $q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}}|\mathbf{x}, \sigma^2 I)$, we can have an analytical solution to $\nabla_{\tilde{\mathbf{x}}}\log q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \frac{1}{\sigma^2}(\mathbf{x} - \tilde{\mathbf{x}})$. This is essentially a direction moving from $\tilde{\mathbf{x}}$ back to \mathbf{x} , and DSM makes the score to match it. Finally, the objective becomes:

$$\begin{aligned} \mathcal{L}_{\text{DSM}} &\approx \frac{1}{2N}\sum_{i=1}^N\left[\left\|\frac{\tilde{\mathbf{x}}_i - \mathbf{x}_i}{\sigma^2} + s_\theta(\tilde{\mathbf{x}}_i)\right\|^2\right], \\ &= \frac{1}{2}\mathbb{E}_{q_\sigma(\mathbf{x},\tilde{\mathbf{x}})}\left[\left\|\frac{\mathbf{x} - \tilde{\mathbf{x}}}{\sigma^2} - s_\theta(\tilde{\mathbf{x}})\right\|^2\right] \\ &= \frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{q_\sigma(\tilde{\mathbf{x}}|\mathbf{x})}\left[\left\|\frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma^2} + s_\theta(\tilde{\mathbf{x}})\right\|^2\right]. \end{aligned} \quad (\text{C.3.6})$$

Additionally, [248] also proves that DSM is equivalent to ESM. Though there exists certain drawbacks [220], DSM serves as a promising tool to enable the SM family as a more applicable solution to EBM.

Noise Conditional Score Network (NCSN). Recently, [218] finds that perturbing data with random Gaussian noise makes the data distribution more powerful than SM model. Thus it proposes Noise Conditional Score Network (NCSN) that can perturb the data using various levels of noise and estimates scores at all levels simultaneously. More concretely, NCSN chooses the Gaussian kernel as noise distribution, *i.e.*, $q_\sigma(\tilde{\mathbf{x}}|\mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}}|\mathbf{x}, \sigma^2 I)$. With L levels of noises, it extends Equation (C.3.6) as the following new objective:

$$\begin{aligned} \ell(\theta; \sigma_i) &= \frac{1}{2}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{q_{\sigma_i}(\tilde{\mathbf{x}}|\mathbf{x})}\left[\left\|\frac{\tilde{\mathbf{x}} - \mathbf{x}}{\sigma_i^2} + s_\theta(\tilde{\mathbf{x}})\right\|^2\right] \\ \mathcal{L}_{\text{NCSN}} &= \frac{1}{L}\sum_{l=1}^L\lambda(\sigma_l)\ell(\theta; \sigma_l), \end{aligned} \quad (\text{C.3.7})$$

where $\lambda(\sigma_i) > 0$ is a coefficient function on σ_i .

Sampling for SM. For the SM family (including ESM, ISM, DSM and NCSN), once we have the score, we can sample the data by a MCMC sampling method called Langevin dynamics:

$$\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{x}}_t + \frac{\epsilon^2}{2} \nabla_{\mathbf{x}} \log p_{\theta}(\mathbf{x}_T) + \epsilon z_t = \tilde{\mathbf{x}}_t + \frac{\epsilon^2}{2} s(\mathbf{x}_t) + \epsilon z_t. \quad (\text{C.3.8})$$

Discussion. Till now, the SM family provides a unique solution for the generative task. It may seem likelihood-free, but recently, another track on diffusion model found that indeed these two research lines can contribute to the same formulation [221]. The only difference is that the diffusion model starts with a variational approximation perspective.

C.3.3. Proof of DSM

Proof of Equation (C.3.5).

First to put this into the ESM, we can have:

$$\begin{aligned} D_F(q(\tilde{\mathbf{x}})||p_{\theta}(\tilde{\mathbf{x}})) &= \frac{1}{2} \mathbb{E}_{q(\tilde{\mathbf{x}})} \|\nabla_{\tilde{\mathbf{x}}} \log p_{\theta}(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}} \log q(\tilde{\mathbf{x}})\|_2^2 \\ &= \frac{1}{2} \mathbb{E}_{q(\tilde{\mathbf{x}})} \left[\|\nabla_{\tilde{\mathbf{x}}} \log p_{\theta}(\tilde{\mathbf{x}})\|_2^2 + 2 \cdot \langle \nabla_{\tilde{\mathbf{x}}} \log p_{\theta}(\tilde{\mathbf{x}}), \nabla_{\tilde{\mathbf{x}}} \log q(\tilde{\mathbf{x}}) \rangle + \|\nabla_{\tilde{\mathbf{x}}} \log q(\tilde{\mathbf{x}})\|_2^2 \right] \\ &= \frac{1}{2} \mathbb{E}_{q(\tilde{\mathbf{x}})} \left[\|s_{\theta}(\tilde{\mathbf{x}})\|_2^2 + 2 \cdot \langle s_{\theta}(\tilde{\mathbf{x}}), \nabla_{\tilde{\mathbf{x}}} \log q(\tilde{\mathbf{x}}) \rangle \right] + C_1, \end{aligned} \quad (\text{C.3.9})$$

where $C_1 = \frac{1}{2} \mathbb{E}_{q(\tilde{\mathbf{x}})} \|\nabla_{\tilde{\mathbf{x}}} \log q(\tilde{\mathbf{x}})\|_2^2$ is a constant and does not depend on the model parameter θ .

Then let's take out the second term, and we can have following:

$$\begin{aligned} &\mathbb{E}_{q(\tilde{\mathbf{x}})} \left[\langle \nabla_{\tilde{\mathbf{x}}} \log q(\tilde{\mathbf{x}}), s_{\theta}(\tilde{\mathbf{x}}) \rangle \right] \\ &= \int_{\tilde{\mathbf{x}}} q(\tilde{\mathbf{x}}) \langle \nabla_{\tilde{\mathbf{x}}} \log q(\tilde{\mathbf{x}}), s_{\theta}(\tilde{\mathbf{x}}) \rangle d\tilde{\mathbf{x}} \\ &= \int_{\tilde{\mathbf{x}}} \langle \nabla_{\tilde{\mathbf{x}}} q(\tilde{\mathbf{x}}), s_{\theta}(\tilde{\mathbf{x}}) \rangle d\tilde{\mathbf{x}} \\ &= \int_{\tilde{\mathbf{x}}} \langle \nabla_{\tilde{\mathbf{x}}} \int_{\mathbf{x}} q(\mathbf{x}) \cdot q(\tilde{\mathbf{x}}|x) dx, s_{\theta}(\tilde{\mathbf{x}}) \rangle d\tilde{\mathbf{x}} \\ &= \int_{\tilde{\mathbf{x}}} \int_{\mathbf{x}} q(\mathbf{x}) \cdot \langle \nabla_{\tilde{\mathbf{x}}} q(\tilde{\mathbf{x}}|x), s_{\theta}(\tilde{\mathbf{x}}) \rangle dx d\tilde{\mathbf{x}} \\ &= \int_{\tilde{\mathbf{x}}} \int_{\mathbf{x}} q(\mathbf{x}) \cdot \langle q(\tilde{\mathbf{x}}|x) \nabla_{\tilde{\mathbf{x}}} \log q(\tilde{\mathbf{x}}|x), s_{\theta}(\tilde{\mathbf{x}}) \rangle dx d\tilde{\mathbf{x}} \\ &= \int_{\tilde{\mathbf{x}}} \int_{\mathbf{x}} q(\mathbf{x}) q(\tilde{\mathbf{x}}|x) \cdot \langle \nabla_{\tilde{\mathbf{x}}} \log q(\tilde{\mathbf{x}}|x), s_{\theta}(\tilde{\mathbf{x}}) \rangle dx d\tilde{\mathbf{x}} \\ &= \mathbb{E}_{q(x, \tilde{\mathbf{x}})} \left[\langle \nabla_{\tilde{\mathbf{x}}} \log q(\tilde{\mathbf{x}}|x), s_{\theta}(\tilde{\mathbf{x}}) \rangle \right] \end{aligned} \quad (\text{C.3.10})$$

So we can put this back to Equation (C.3.9) and let ESM be as:

$$\begin{aligned} D_F(q(\tilde{\mathbf{x}})||p_{\theta}(\tilde{\mathbf{x}})) &= \frac{1}{2} \mathbb{E}_{q(\tilde{\mathbf{x}})} \left[\|s_{\theta}(\tilde{\mathbf{x}})\|_2^2 + 2 \cdot \langle s_{\theta}(\tilde{\mathbf{x}}), \nabla_{\tilde{\mathbf{x}}} \log q(\tilde{\mathbf{x}}) \rangle \right] + C_1 \\ &= \frac{1}{2} \mathbb{E}_{q(x, \tilde{\mathbf{x}})} \left[\|s_{\theta}(\tilde{\mathbf{x}})\|_2^2 \right] + \mathbb{E}_{q(x, \tilde{\mathbf{x}})} \left[\langle \nabla_{\tilde{\mathbf{x}}} \log q(\tilde{\mathbf{x}}|x), s_{\theta}(\tilde{\mathbf{x}}) \rangle \right] + C_1. \end{aligned} \quad (\text{C.3.11})$$

And then we can get the following equivalent objective by some special reconstruction:

$$\begin{aligned} D_F(q(\tilde{\mathbf{x}})||p_\theta(\tilde{\mathbf{x}})) &= \frac{1}{2}\mathbb{E}_{q(x,\tilde{\mathbf{x}})}\left[\|s_\theta(\tilde{\mathbf{x}})\|^2\right] + \mathbb{E}_{q(x,\tilde{\mathbf{x}})}\left[\langle\nabla_{\tilde{\mathbf{x}}}\log q(\tilde{\mathbf{x}}|x), s_\theta(\tilde{\mathbf{x}})\rangle\right] + C_2 + \Delta \\ &= \frac{1}{2}\mathbb{E}_{q(x,\tilde{\mathbf{x}})}\left[\|s_\theta(\tilde{\mathbf{x}}) - \nabla_{\tilde{\mathbf{x}}}\log q(\tilde{\mathbf{x}}|x)\|^2\right] + \Delta. \end{aligned} \quad (\text{C.3.12})$$

where $C_2 = \frac{1}{2}\mathbb{E}_{q(\tilde{\mathbf{x}})}\|\nabla_{\tilde{\mathbf{x}}}\log q(\tilde{\mathbf{x}}|x)\|^2$ is a re-constructed constant.

End of proof.

C.4. Diffusion Model

Another generative modeling track is the denoising diffusion probabilistic model (DDPM) [92, 216]. The diffusion model is composed of two processes: a forward process that adds noise to the data and a backward process that does denoising to generate the true data. Below we give a brief summary of the Gaussian diffusion model introduced in [92].

C.4.1. Pipeline of Denoising Diffusion Probabilistic Model

Forward process. Give a data point from the real distribution $\mathbf{x}_0 \sim q(\mathbf{x})$, the forward diffusion process is that we add small amount of Gaussian noise to the sample in T steps, producing a sequence of noisy samples $\mathbf{x}_1, \dots, \mathbf{x}_T$. The step sizes are controlled by a variance schedule $\{\beta_t \in (0, 1)\}_{t=1}^T$:

$$q(\mathbf{x}_T|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_T; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t I), \quad (\text{C.4.1})$$

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_T|\mathbf{x}_{t-1}). \quad (\text{C.4.2})$$

A nice property of the forward process is that we can sample \mathbf{x}_t at any arbitrary timestep t in a closed form using the reparameterization trick. Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$, we have:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1 - \bar{\alpha}_t)I). \quad (\text{C.4.3})$$

Then using the Bayes theorem, $q(\mathbf{x}_{t-1}|\mathbf{x}_T, \mathbf{x}_0)$ can be written as a Gaussian:

$$\begin{aligned} q(\mathbf{x}_{t-1}|\mathbf{x}_T, \mathbf{x}_0) &= \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}(\mathbf{x}_T, \mathbf{x}_0), \tilde{\beta}_t I) \\ &= \mathcal{N}(\mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_T - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}}z_t), \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t}\beta_t) \end{aligned} \quad (\text{C.4.4})$$

Reverse process. Under a reasonable setting for β_t and T [46], the distribution $q(\mathbf{x}_T)$ is nearly an isotropic Gaussian, and sampling \mathbf{x}_T is trivial. Then for the reverse process, we need $q(\mathbf{x}_{t-1}|\mathbf{x}_T)$. [216] claims that as $T \rightarrow \infty$ and $\beta_t \rightarrow 0$, $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ approaches a diagonal Gaussian distribution. To this end, it is sufficient to train a neural network to predict a mean μ_θ and a diagonal covariance matrix Σ_θ :

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_T) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_T, t), \Sigma_\theta(\mathbf{x}_T, t)). \quad (\text{C.4.5})$$

Parameterization and variational lower bound. The hidden variables are $x_{1:T}$, and inference is to infer the latent variables, *i.e.*, $p(x_{1:T}|\mathbf{x}_0)$. Variational inference is to use $p(x_{1:T}|\mathbf{x}_0)$ to estimate the true posterior $q(x_{1:T}|\mathbf{x}_0)$. If we use $x_{1:T}$ as z , and \mathbf{x}_0 as x , then it is resemble to the VAE. Recall that

$$\begin{aligned}
KL(q(z|x)||p(z|x)) &= \mathbb{E}_{q(z|x)} \left[\log \frac{q(z|x)}{p(z|x)} \right] \\
&= \mathbb{E}_{q(z|x)} \left[\log \frac{q(z|x)p(x)}{p(x,z)} \right] \\
&= \log p(x) + \mathbb{E}_{q(z|x)} \left[\log \frac{q(z|x)}{p(x,z)} \right].
\end{aligned} \tag{C.4.6}$$

To adapt this to the diffusion model setting, we can have:

$$KL(q(x_{1:T}|\mathbf{x}_0)||p(x_{1:T}|\mathbf{x}_0)) = \log p(\mathbf{x}_0) + \mathbb{E}_{q(x_{1:T}|\mathbf{x}_0)} \left[\log \frac{q(x_{1:T}|\mathbf{x}_0)}{p(x_{0:T})} \right], \tag{C.4.7}$$

and our goal becomes to maximize the variational lower bound (VLB):

$$\begin{aligned}
\mathcal{L}_{VLB} &= \mathbb{E}_q \left[\log \frac{q(x_{1:T}|\mathbf{x}_0)}{p_\theta(x_{0:T})} \right] \\
&= \mathbb{E}_q \left[\log \frac{\prod_{t=1}^T q(\mathbf{x}_T|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_T)} \right] \\
&= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_T|\mathbf{x}_{t-1})}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_T)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \\
&= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_T|\mathbf{x}_{t-1}, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_T)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \\
&= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_T, \mathbf{x}_0) \cdot q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_T) \cdot q(\mathbf{x}_{t-1}|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \quad \text{Baye's rule} \\
&= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_T, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_T)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_1|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \\
&= \mathbb{E}_q \left[-\log p_\theta(\mathbf{x}_T) + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_T, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_0)} + \log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_0|\mathbf{x}_1)} \right] \\
&= \mathbb{E}_q \left[\log \frac{q(\mathbf{x}_T|\mathbf{x}_0)}{p_\theta(\mathbf{x}_T)} + \sum_{t=2}^T \log \frac{q(\mathbf{x}_{t-1}|\mathbf{x}_T, \mathbf{x}_0)}{q(\mathbf{x}_{t-1}|\mathbf{x}_0)} - \log p_\theta(\mathbf{x}_0|\mathbf{x}_1) \right] \\
&= KL[q(\mathbf{x}_T|\mathbf{x}_0)||p_\theta(\mathbf{x}_T)] + \sum_{t=2}^T KL[q(\mathbf{x}_{t-1}|\mathbf{x}_T, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_T)] - \mathbb{E}_q[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)] \\
&= \underbrace{KL[q(\mathbf{x}_T|\mathbf{x}_0)||p_\theta(\mathbf{x}_T)]}_{\mathcal{L}_T} + \sum_{t=2}^T \underbrace{KL[q(\mathbf{x}_{t-1}|\mathbf{x}_T, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_T)]}_{\mathcal{L}_{t-1}} - \underbrace{\mathbb{E}_q[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)]}_{\mathcal{L}_0}
\end{aligned} \tag{C.4.8}$$

Thus, we want to model $q(\mathbf{x}_{t-1}|\mathbf{x}_T, \mathbf{x}_0)$ with parameterization $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_T)$. According to Equation (C.4.4), we can have:

$$\begin{aligned}
\mathcal{L}_t &= \mathbb{E}_q \left[\log \frac{q(\mathbf{x}_{t-1} | \mathbf{x}_T, \mathbf{x}_0)}{p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_T)} \right] \\
&= \mathbb{E}_{\mathbf{x}_0, z} \left[-\frac{1}{2\|\Sigma_\theta\|^2} \|\tilde{\mu}_t(\mathbf{x}_T, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_T, t)\|^2 \right] \\
&= \mathbb{E}_{\mathbf{x}_0, z} \left[-\frac{1}{2\|\Sigma_\theta\|^2} \cdot \frac{1}{\alpha_t} \cdot \left\| \mathbf{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_t - \mathbf{x}_T + \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} z_\theta(\mathbf{x}_T, t) \right\|^2 \right] \\
&= \mathbb{E}_{\mathbf{x}_0, z} \left[-\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)\|\Sigma_\theta\|^2} \cdot \|z_t - z_\theta(\mathbf{x}_T, t)\|^2 \right] \\
&= \mathbb{E}_{\mathbf{x}_0, z} \left[-\frac{\beta_t^2}{2\alpha_t(1-\bar{\alpha}_t)\|\Sigma_\theta\|^2} \cdot \|z_t - z_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}z_t, t)\|^2 \right],
\end{aligned} \tag{C.4.9}$$

Simplification. There is a nice strategy proposed in [92]: the objective function in Equation (C.4.9) can be simplified by ignoring the weighting term:

$$\mathcal{L}_t^{simple} = \mathbb{E}_{\mathbf{x}_0, z} \left[\|z_t - z_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}z_t, t)\|^2 \right] \tag{C.4.10}$$

C.4.2. Important Tricks

The DDPM [92] also adopts the following tricks in the training and inference.

- $\Sigma_\theta(\mathbf{x}_T, t) = \sigma_t^2 I$. Then DDPM empirically tests $\sigma_t^2 = \beta_t$ and $\sigma_t^2 = \tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \beta_t$. Both have similar results. And these two are the two extreme choices corresponding to the upper and lower bounds on reverse process entropy with coordinatewise unit variance [216].
- The second trick is that we model $p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_T) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_T, t); \sigma_t^2 I)$. Then the loss term becomes:

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \|\tilde{\mu}(\mathbf{x}_T, \mathbf{x}_0) - \mu_\theta(\mathbf{x}_T, t)\|^2 \right] \tag{C.4.11}$$

So one straightforward way is to directly model the mean, *i.e.*, to match μ_θ and $\tilde{\mu}$.

- Meanwhile, during the diffusion process, we can have \mathbf{x}_0 and \mathbf{x}_T . Thus, we can have $\tilde{\mu}(\mathbf{x}_T, \mathbf{x}_0)$ as a function of (\mathbf{x}_T, ϵ) or (\mathbf{x}_0, ϵ) . In specific, we can write it as:

$$L_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon \right) - \mu_\theta(\mathbf{x}_T, t) \right\|^2 \right] \tag{C.4.12}$$

Since \mathbf{x}_T can be obtained by \mathbf{x}_0 , then we may as well model $\mu_\theta(\mathbf{x}_T, t) = \tilde{\mu}_t(\mathbf{x}_T, \mathbf{x}_0(\mathbf{x}_T, \epsilon_t)) = \frac{1}{\sqrt{\alpha_t}} (\mathbf{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_T, t))$, and the objective function becomes:

$$\begin{aligned}
L_{t-1} &= \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \left\| \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon \right) - \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_T, t) \right) \right\|^2 \right] \\
&= \mathbb{E}_q \left[\frac{\beta_t^2}{2\sigma_t^2 \alpha_t (1-\bar{\alpha}_t)} \|\epsilon - \epsilon_\theta(\mathbf{x}_T, t)\|^2 \right]
\end{aligned} \tag{C.4.13}$$

- Thus, during sampling, the mean is

$$\mu_\theta(\mathbf{x}_T, t) = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\mathbf{x}_T, t) \right), \tag{C.4.14}$$

thus the sampling is obtained by:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_T - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_T, t)) + \sqrt{\beta_t}\epsilon \quad //\text{DDPM's paper} \quad (\text{C.4.15})$$

- Further, if we want to model the score, *i.e.*, the $\log_{\tilde{x}} \log q(\tilde{x}|x)$, and then the score network defined here needs to have a shift:

$$\begin{aligned} \log_{\tilde{x}} q(\tilde{x}|x) &= \log_{\mathbf{x}_T} q(\mathbf{x}_T|\mathbf{x}_0) \\ &= -\log_{\mathbf{x}_T} \mathcal{N}(\mathbf{x}_T; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)I) \\ &= -\frac{\sqrt{\bar{\alpha}_t}\mathbf{x}_0}{\sqrt{1-\bar{\alpha}_t}} \\ &= \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_T + \beta_t\epsilon_\theta(\mathbf{x}_T, t)) + \sqrt{\beta_t}\epsilon. \end{aligned} \quad (\text{C.4.16})$$

C.5. Stochastic Differential Equation

A more recent work [221] unifies the score matching and DDPM into a unified framework, the stochastic differential equation (SDE). First let's do a quick recap on the NCSN and DDPM.

C.5.1. Review of NCSN and DDPM

NCSN. The objective function is:

$$\mathcal{L} = \sum_{t=1}^T \sigma_t^2 \cdot \mathbb{E}_{p_{data}(x)} \mathbb{E}_{q_{\sigma_t}(\mathbf{x}_T|x)} \left[\left\| s_\theta(\mathbf{x}_T, \sigma_t) - \nabla_{\mathbf{x}_T} \log q_{\sigma_t}(\tilde{x}|x) \right\|^2 \right]. \quad (\text{C.5.1})$$

There is no notion of forward and backward, and the **sampling** is achieved by the Langevine dynamics:

$$\mathbf{x}_T^m = \mathbf{x}_T^{m-1} + \delta s_\theta(\mathbf{x}_T^{m-1}, \sigma_t) + \sqrt{2\delta}\epsilon, \quad m = 1, \dots, M, \quad (\text{C.5.2})$$

where δ is the step size and $\epsilon \sim \mathcal{N}(0, I)$. The above is repeated with:

- From $t = T$ to $t = 1$.
- $\mathbf{x}_T^0 \sim \mathcal{N}(0, \sigma_T^2 I)$ for $t = T$.
- $\mathbf{x}_T^0 = x_{t+1}^M$ when $t < T$.

DDPM. The **forward** (non-modeling) is:

$$p(\mathbf{x}_T|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_T; \sqrt{\alpha_t}\mathbf{x}_{t-1}, (1-\alpha_t)I). \quad (\text{C.5.3})$$

The **backward** (modeling part) is:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_T) = \mathcal{N}\left(\mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_T + \beta_t s_\theta(\mathbf{x}_T, t)), \beta_t I\right) \quad (\text{C.5.4})$$

The **objective** function is:

$$\mathcal{L} = \sum_{t=1}^T \mathbb{E}_{p_{data}(x_0)} \mathbb{E}_{p_{\alpha_t}(\mathbf{x}_T|x_0)} \left[\left\| s_\theta(\mathbf{x}_T, t) - \nabla_{\mathbf{x}_T} \log q(\mathbf{x}_T|\mathbf{x}_0) \right\|^2 \right] \quad (\text{C.5.5})$$

The **sampling** is as follows:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}(\mathbf{x}_T + \beta_t s_\theta(\mathbf{x}_T, t)) + \sqrt{\beta_t} \epsilon_t, \quad t = T, \dots, 1, \quad (\text{C.5.6})$$

where $\mathbf{x}_T \sim \mathcal{N}(0, I)$ and $\epsilon_t \sim \mathcal{N}(0, I)$. This is called the ancestral sampling since it amounts to performing ancestral sampling from the graphical mode [221].

Comparison of NCSN and DDPM. Note that in DDPM, it is modeling $KL(q(\mathbf{x}_{t-1}|\mathbf{x}_T, \mathbf{x}_0)||p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_T))$, while NCSN is modeling $s_\theta(\mathbf{x}_T, t) - \nabla_{\mathbf{x}_T} \log p(\mathbf{x}_{t-1}|\mathbf{x}_T)$ directly. Essentially, these two are equivalent, because:

$$-\sqrt{1 - \bar{\alpha}_t} s_\theta(\mathbf{x}_T, t) = \epsilon_\theta(\mathbf{x}_T, t). \quad (\text{C.5.7})$$

C.5.2. Stochastic Differential Equation

Then we introduce the how NCSN and DDPM are solutions to Stochastic Differential Equation (SDE). The SDE is also formulated with the forward and backward processes.

Forward process is

$$d\mathbf{x} = f(\mathbf{x}, t)dt + g(t)dw, \quad (\text{C.5.8})$$

where $f(\mathbf{x}, t)$ is the vector-value drift coefficient, $g(t)$ is the diffusion coefficient, and dw is the Wiener process.

Backward process is:

$$d\mathbf{x} = [f(\mathbf{x}, t) - g(t)^2 \nabla_{\mathbf{x}} \log p(\mathbf{x}_T)]dt + g(t)dw \quad (\text{C.5.9})$$

Then the questions is how to estimate the score $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$.

C.5.3. Stochastic Differential Equation and Score Matching

According to [221], the objective of solutions to SDE can be written in the form of score matching:

$$\mathcal{L} = \mathbb{E}_t \mathbb{E}_{\mathbf{x}_0} \mathbb{E}_{\mathbf{x}_T | \mathbf{x}_0} \left[\lambda(t) \left\| s_\theta(\mathbf{x}_T, t) - \nabla_{\mathbf{x}_T} \log p(\mathbf{x}_T | \mathbf{x}_0) \right\|^2 \right], \quad (\text{C.5.10})$$

where $\lambda(t)$ is a weighting function. With sufficient data and model capacity, the optimal $s_\theta(\mathbf{x}_T, t)$ equals to $\nabla_{\mathbf{x}} \log p(\mathbf{x}_T)$ for almost all \mathbf{x}_T and t . Then we will review how to match the NCSN and DDPM into this framework.

NCSN and VE SDE. The discretization of VE SDE yields NCSN. The **forward** process is:

$$\begin{aligned} \mathbf{x}_T &= \mathbf{x}_{t-1} + \sqrt{\sigma_t^2 - \sigma_{t-1}^2} \epsilon_{t-1}, & // \text{ discrete Markov chain} \\ dx &= \sqrt{\frac{d[\sigma^2(t)]}{dt}} dw, & // \text{ continuous SDE} \end{aligned} \quad (\text{C.5.11})$$

Assumption: the $\{\sigma_t\}, t = 1, 2, \dots, T$ is a geometric sequence. Then the transition kernel becomes:

$$dx = \sigma_{min} \left(\frac{\sigma_{max}}{\sigma_{min}} \right)^2 \sqrt{2 \log \frac{\sigma_{max}}{\sigma_{min}}} dw \quad (\text{C.5.12})$$

and the perturbation kernel can be obtained by

$$p(\mathbf{x}_T|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_T; \mathbf{x}_0, \sigma_{min}^2 \left(\frac{\sigma_{max}}{\sigma_{min}}\right)^{2t} I\right) \quad (\text{C.5.13})$$

This always give a process with exploding variance, so this is called VE SDE.

DDPM and VP SDE The forward process is:

$$\mathbf{x}_T = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1 - \alpha_t}\epsilon_{t-1}, \quad // \text{ discrete Markov chain} \quad (\text{C.5.14})$$

$$d\mathbf{x} = -\frac{1}{2}(1 - \alpha_t)\mathbf{x}dt + \sqrt{\beta(t)}d\mathbf{w}, \quad // \text{ continuous SDE}$$

If we use the arithmetic sequence for $\{\beta\}_{t=1}^T$, the transition kernel for VP SDE is:

$$d\mathbf{x} = -\frac{1}{2}(\beta_{min} + t(\beta_{max} - \beta_{min}))\mathbf{x}dt + \sqrt{\beta_{min} + t(\beta_{max} - \beta_{min})}d\mathbf{w} \quad (\text{C.5.15})$$

and the perturbation kernel is:

$$p(\mathbf{x}_T|\mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_T; e^{-\frac{1}{4}t^2(\beta_{max}-\beta_{min}-\frac{1}{2}t\beta_{min})}\mathbf{x}_0, I - Ie^{-\frac{1}{2}t^2(\beta_{max}-\beta_{min}-t\beta_{min})}\right) \quad (\text{C.5.16})$$

The drift coefficient is $-\frac{1}{2}(\beta_{min} + t(\beta_{max} - \beta_{min}))\mathbf{x}$, and the diffusion coefficient is $\beta_{min} + t(\beta_{max} - \beta_{min})$.

C.6. Mutual Information and Equivalent Conditional Likelihoods

To maximize the mutual information between variable X, Y is equivalent to optimize the following equation:

$$\mathcal{L} = \frac{1}{2}\mathbb{E}_{p(\mathbf{x}, \mathbf{y})}[\log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x}|\mathbf{y})]. \quad (\text{C.6.1})$$

Proof. First we can get a lower bound of MI. Assuming that there exist (possibly negative) constants a and b such that $a \leq H(X)$ and $b \leq H(Y)$, *i.e.*, the lower bounds to the (differential) entropies, then we have:

$$\begin{aligned} I(X; Y) &= \frac{1}{2}(H(X) + H(Y) - H(Y|X) - H(X|Y)) \\ &\geq \frac{1}{2}(a + b - H(Y|X) - H(X|Y)) \\ &\geq \frac{1}{2}(a + b) + \mathcal{L}, \end{aligned} \quad (\text{C.6.2})$$

where the loss \mathcal{L} is defined as:

$$\begin{aligned} \mathcal{L} &= \frac{1}{2}(-H(Y|X) - H(X|Y)) \\ &= \frac{1}{2}\mathbb{E}_{p(\mathbf{x}, \mathbf{y})}[\log p(\mathbf{x}|\mathbf{y})] + \frac{1}{2}\mathbb{E}_{p(\mathbf{x}, \mathbf{y})}[\log p(\mathbf{y}|\mathbf{x})]. \end{aligned} \quad (\text{C.6.3})$$

End of proof.

Empirically, we use energy-based models to model the distributions. The condition on the existence of a and b can be understood as the requirements that the two distributions (p_x, p_y) are not collapsed.

C.6.1. Variational Representation Reconstruction

The variational representation reconstruction (VRR) was first introduced in GraphMVP [153]. There are two mirroring terms in Equation (C.6.1), and here we take one term for illustration. The goal of VRR is to take the variational lower bound to maximize:

$$\mathbb{E}_{p(\mathbf{x}, \mathbf{y})} [\log p(\mathbf{y}|\mathbf{x})]. \quad (\text{C.6.4})$$

The objective function to Equation (C.6.4) has a variational lower bound as:

$$\log p(\mathbf{y}|\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}_x|\mathbf{x})} [\log p(\mathbf{y}|\mathbf{z}_x)] - KL(q(\mathbf{z}_x|\mathbf{x})||p(\mathbf{z}_x)). \quad (\text{C.6.5})$$

And VRR proposes a proxy solution by doing the reconstruction on the representation space instead of the data space:

$$\mathcal{L}_G = \mathcal{L}_{\text{VRR}} = \mathbb{E}_{q(\mathbf{z}_x|\mathbf{x})} [\|q(\mathbf{z}_x) - \text{SG}(h_{\mathbf{y}})\|^2] + \beta KL(q(\mathbf{z}_x|\mathbf{x})||p(\mathbf{z}_x)). \quad (\text{C.6.6})$$

C.7. Implementation Details of MoleculeSDE

In this section, we illustrate the details of our proposed MoleculeSDE, including the featurization, backbone models, hyperparameters, architectures for the score networks, etc. The solution in Appendix C.6.1 to Equation (C.6.1) is indeed a conditional generative method to solving the self-supervised learning (SSL). Meanwhile, it is only a proxy solution by conducting the reconstruction on the representation space. Thus, we want to explore a more accurate and explicit estimation to the generative reconstruction on the data space (*i.e.*, the 2D topology and 3D geometry of molecules).

C.7.1. Backbone Models

For the backbone models, we stick with the existing pretraining works [99, 145, 153, 255], which can better illustrate the effectiveness of our proposed methods. We use Graph Isomorphism Network (GIN) [266] for modeling the 2D topology and SchNet [207] for 3D conformation, respectively.

C.7.2. Molecule Featurization

The molecule featurization is an essential factor that should be taken into consideration. A recent work [230] has empirically verified that utilizing the rich atom feature. We follow this strategy and employ the featurization from MoleculeNet [263] and OGB [97]. In specific, we have the atom and bond featurization in Table 44.

Table 44. Featurization for atoms and bonds.

	Hyperparameter	Value
Atom Featurization	Atom Type	[0, 118]
	Atom Chirality	{unspecified, unrecognized type, tetrahedral with clockwise rotation, tetrahedral: counter-clockwise rotation}
	Atom Degree	[0, 10]
	Formal Charge	[-5, 5]
	Number of Hydrogen	[0, 8]
	Number of Unpaired Electrons	[0, 4]
	Hybridization	{SP, SP2, SP3, SP3D, SP3D2}
	Is Aromatic	{False, True}
Bond Featurization	Is In Ring	{False, True}
	Bond Type	{single, double, triple, aromatic}
	Bond Stereotype	{none, Z variant, E variant, Cis, Trans, any}
	Is conjugated	{False, True}

We also want to highlight that such an atom featurization is only available for the topological graph; while for 3D conformation, only the atom type information is available. The other atom information requires either the topology information (*e.g.*, degree, number of Hydrogen) or chemical rules (*e.g.*, chirality) to obtain, and they have not been utilized for the molecule geometric modeling [207].

C.7.3. Pretraining Hyperparameters

The pretraining pipeline is shown in Figure 6 and the objective function is Section 4.4. Below in Table 45, we illustrate the key hyperparameters used in MoleculeSDE.

Table 45. Hyperparameter specifications for MoleculeSDE.

Hyperparameter	Value
epochs	{50, 100}
learning rate 2D GNN	{1e-5, 1e-6}
learning rate 3D GNN	{1e-5, 1e-6}
SDE option	{VE, VP}
masking ratio M	{0, 0.3}
β	{[0.1, 10]}
number of steps	{1000}
α_1	{0, 1}
α_2	{0}
α_3	{0}

C.7.4. SE(3)-Equivariant SDE Model: From Topology to Conformation

We list the detailed structure of the SE(3)-equivariant SDE model in Figure 23.

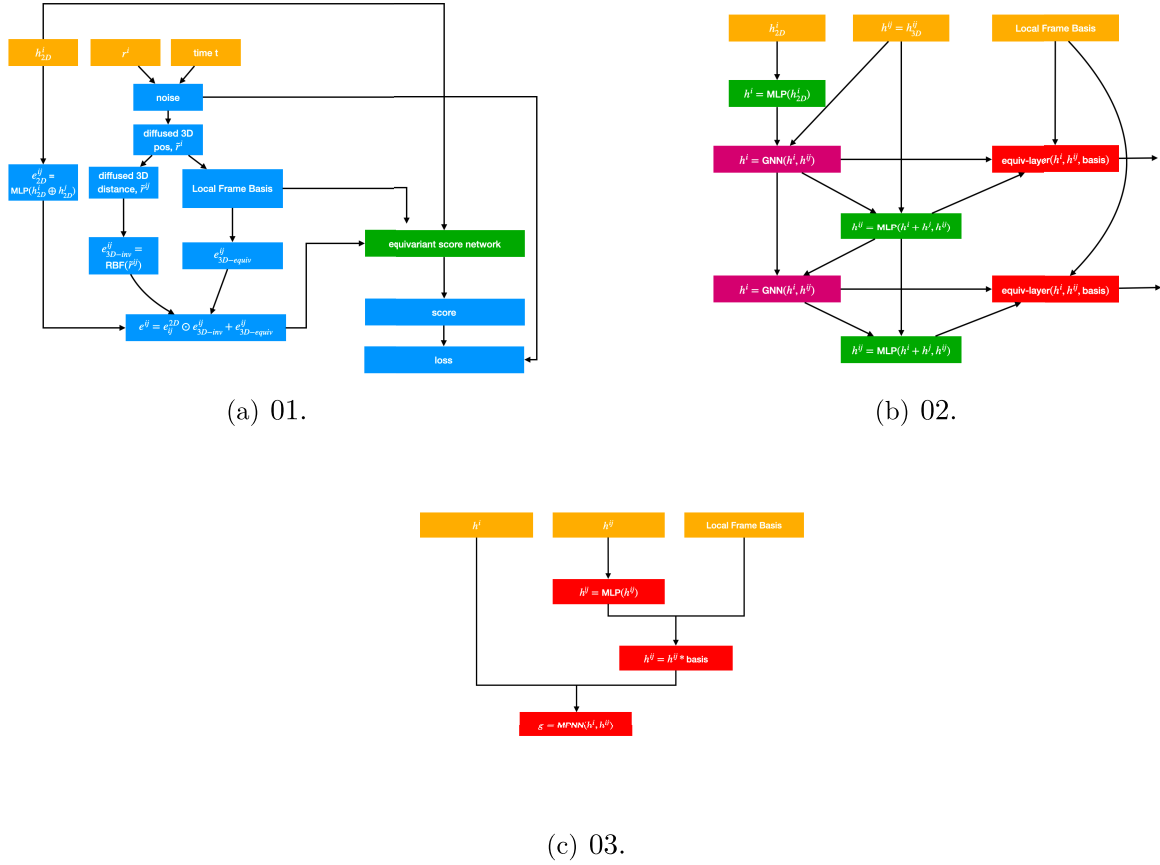


Figure 23. Pipeline for the SE(3)-equivariant SDE model from topology to conformation.

C.7.5. SE(3)-Invariant SDE Model: From Conformation to Topology

We list the detailed structure of the SE(3)-invariant SDE model in Figure 24.

Similarly with the $2D \rightarrow 3D$ procedure, we first merge the 3D representation \mathbf{y} with the diffused atom feature \mathbf{X}_t :

$$H_0 = \text{MLP}(\mathbf{X}_t) + \mathbf{y}.$$

Since the noised \mathbf{E}_t becomes a dense adjacency matrix, we will follow implement a densed GCN to model $\nabla_{\mathbf{X}_t} \log p_t(\mathbf{x}_t|\mathbf{y})$:

$$S_\theta^{\mathbf{X}_t}(\mathbf{x}_t) = \text{MLP}(\text{Concat}\{H_0 || \dots || H_L\}),$$

where $H_{i+1} = \text{GCN}(H_i, \mathbf{E}_t)$. On the other hand, $\nabla_{\mathbf{E}_t} \log p_t(\mathbf{x}_t|\mathbf{y}) \in \mathbf{R}^{n \times n}$ is modeled by an unnormalized dot product attention (without softmax):

$$S_\theta^{\mathbf{E}_t}(\mathbf{x}_t) = \text{MLP}(\{\text{Attention}(H_i)\}_{0 \leq i \leq L}).$$

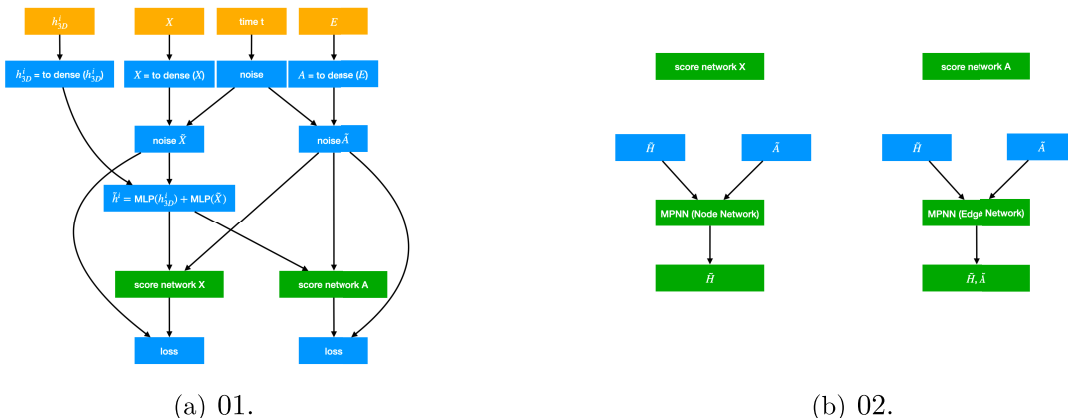


Figure 24. Pipeline for the SE(3)-invariant SDE model from conformation to topology.

C.8. Ablation Studies

This section provides more ablation studies to verify key concepts in molecule pretraining.

C.8.1. Ablation Study on Generative SSL Pretraining

We first provide a comprehensive comparison of the effect of the generative SSL part.

Pretraining. In GraphMVP [153], the generative SSL is variational representation reconstruction (VRR). In MoleculeSDE, the generative SSL is composed of two SDE models.

Downstream. We consider both the 2D and 3D downstream tasks. We can tell that the generative SSL (SDE) in MoleculeSDE is better than the generative SSL (VRR) in GraphMVP by a large margin on 27 out of 28 tasks.

Table 46. Ablation studies on generative SSL comparison. Results for molecular property prediction tasks (with 2D topology only). The best results are marked in **bold** and **bold**, respectively.

Pre-training	BBBP \uparrow	Tox21 \uparrow	ToxCast \uparrow	Sider \uparrow	ClinTox \uparrow	MUV \uparrow	HIV \uparrow	Bace \uparrow	Avg \uparrow
VRR (GraphMVP)	62.4 \pm 1.71	73.6 \pm 1.09	61.4 \pm 0.56	57.2 \pm 1.11	86.5\pm3.02	75.5 \pm 1.58	75.4 \pm 0.96	72.7 \pm 2.16	70.61
SDE-VE (MoleculeSDE)	68.8\pm3.53	76.5\pm0.28	64.9\pm0.14	59.2\pm0.44	86.1\pm2.15	77.7\pm2.15	77.0\pm0.66	79.6\pm0.66	73.73
SDE-VP (MoleculeSDE)	65.5\pm3.25	75.6\pm0.36	63.4\pm0.22	59.8\pm0.23	81.1 \pm 1.83	80.1\pm1.10	78.6\pm0.31	79.0\pm0.79	72.89

Table 47. Ablation studies on generative SSL comparison. Results on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for testing. The evaluation is mean absolute error. The best results are marked in **bold** and **bold**, respectively.

Pretraining	Alpha \downarrow	Gap \downarrow	HOMO \downarrow	LUMO \downarrow	Mu \downarrow	Cv \downarrow	G298 \downarrow	H298 \downarrow	R2 \downarrow	U298 \downarrow	U0 \downarrow	Zpve \downarrow
VRR (GraphMVP)	0.058	44.64	27.32	22.50	0.030	0.030	14.96	14.69	0.127	14.35	13.96	1.680
SDE-VE (MoleculeSDE)	0.056	41.84	25.79	21.63	0.027	0.029	11.47	10.71	0.233	11.04	10.95	1.474
SDE-VP (MoleculeSDE)	0.056	42.75	25.84	21.52	0.027	0.029	11.90	11.85	0.200	12.03	11.69	1.453

Table 48. Ablation studies on generative SSL comparison. Results on eight force prediction tasks from MD17. We take 1K for training, 1K for validation, and 48K to 991K molecules for the test concerning different tasks. The evaluation is the mean absolute error. The best results are marked in **bold** and **bold**, respectively.

Pretraining	Aspirin ↓	Benzene ↓	Ethanol ↓	Malonaldehyde ↓	Naphthalene ↓	Salicylic ↓	Toluene ↓	Uracil ↓
VRR (GraphMVP)	1.177	0.389	0.533	0.828	0.562	0.806	0.528	0.717
SDE-VE (MoleculeSDE)	1.247	0.364	0.448	0.735	0.483	0.785	0.480	0.575
SDE-VP (MoleculeSDE)	1.087	0.358	0.300	0.880	0.517	0.788	0.540	0.675

C.8.2. Ablation Study on Atom Features and Comparison with Conformation Generation Methods

As recently discussed in [230], the atom feature plays an important role in molecule modeling, especially for 2D topological modeling. We carefully consider this in our work.

Note that for GIN in Table 10, we are using comprehensive atom and bond features, as shown in Appendix C.7.2. For the ablation study in Table 13, to make it a fair comparison between GIN and SchNet, we further employed merely the atom type (the same as 3D conformation modeling) and the bond type for 2D topology modeling. We name these two as "GIN with rich features" and the GIN in Table 10 as "GIN with simple features", respectively. The results and comparison with conformation generation methods are shown in Table 49.

Table 49. Ablation study on the effect of rich features for GIN and comparison with SchNet on conformation generation (CG) methods.

Model	CG Method	BBBP	Sider	ClinTox	Bace
GIN with rich features	–	68.1±0.59	57.0±1.33	83.7 ±2.93	76.7±2.51
GIN with simple features	–	64.1±1.79	58.4±0.50	63.1±7.21	76.5±2.96
SchNet	MMFF	61.4±0.29	59.4±0.27	64.6±0.50	74.3±0.66
SchNet	ConfGF	62.7±1.97	60.1±0.87	64.1±2.83	73.2±3.53
SchNet	ClofNet	61.7±1.19	56.0±0.10	58.2±0.44	62.5±0.17
SchNet	MoleculeSDE	65.2±0.43	60.5±0.39	72.9±1.02	78.6±0.40

Observation 1. We can tell that using rich or simple features plays an important role in GIN model. This can be observed when comparing the GIN in Table 10 and SchNet in Table 13, and we summarize them in the first two rows in Table 49.

Observation 2. Additionally, we can tell that SchNet on MoleculeSDE can outperform GIN with simple features, showing that in terms of the MoleculeSDE can extract more useful geometric information. Meanwhile, GIN with rich features performs better on two tasks, especially a large margin in ClinTox. This reveals that the heuristic 2D topological information can also convey some information that is missing in MoleculeSDE.

Thus, the main message we want to deliver to the audience is that MoleculeSDE is better in terms of the conformation generation, and can be combined with the 2D topology modeling for future works.

C.8.3. Ablation Study on The Effect of Contrastive Learning in MoleculeSDE

Table 50. Ablation studies on α_1 in MoleculeSDE. Results for molecular property prediction tasks (with 2D topology only). The best results are marked in **bold** for each pair of $\alpha_1 \in \{0, 1\}$.

	α_1	BBBP \uparrow	Tox21 \uparrow	ToxCast \uparrow	Sider \uparrow	ClinTox \uparrow	MUV \uparrow	HIV \uparrow	Bace \uparrow	Avg \uparrow
VE	0	68.8 \pm 3.53	76.5\pm0.28	64.9 \pm 0.14	59.2 \pm 0.44	86.1 \pm 2.15	77.7 \pm 2.15	77.0 \pm 0.66	79.6 \pm 0.66	73.73
	1	73.2\pm0.48	76.5\pm0.33	65.2\pm0.31	59.6\pm0.82	86.6\pm3.73	79.9\pm0.19	78.5\pm0.28	80.4\pm0.92	74.98
VP	0	65.5 \pm 3.25	75.6 \pm 0.36	63.4 \pm 0.22	59.8 \pm 0.23	81.1 \pm 1.83	80.1 \pm 1.10	78.6 \pm 0.31	79.0 \pm 0.79	72.89
	1	71.8\pm0.76	76.8\pm0.34	65.0\pm0.26	60.8\pm0.39	87.0\pm0.53	80.9\pm0.37	78.8\pm0.92	79.5\pm2.17	75.07

Table 51. Ablation studies on α_1 in MoleculeSDE. Results on 12 quantum mechanics prediction tasks from QM9. We take 110K for training, 10K for validation, and 11K for testing. The evaluation is the mean absolute error. The best results are marked in **bold** for each pair of $\alpha_1 \in \{0, 1\}$.

	α_1	Alpha \downarrow	Gap \downarrow	HOMO \downarrow	LUMO \downarrow	Mu \downarrow	Cv \downarrow	G298 \downarrow	H298 \downarrow	R2 \downarrow	U298 \downarrow	U0 \downarrow	Zpve \downarrow
VE	0	0.056	41.84	25.79	21.63	0.027	0.029	11.47	10.71	0.233	11.04	10.95	1.474
	1	0.055	41.88	25.62	21.51	0.026	0.029	12.91	12.37	0.142	12.68	12.56	1.608
VP	0	0.056	42.75	25.84	21.52	0.027	0.029	11.90	11.85	0.200	12.03	11.69	1.453
	1	0.054	41.77	25.74	21.41	0.026	0.028	13.07	12.05	0.151	12.54	12.04	1.587

Table 52. Ablation studies on α_1 in MoleculeSDE. Results on eight force prediction tasks from MD17. We take 1K for training, 1K for validation, and 48K to 991K molecules for the test concerning different tasks. The evaluation is the mean absolute error. The best results are marked in **bold** for each pair of $\alpha_1 \in \{0, 1\}$.

	α_1	Aspirin \downarrow	Benzene \downarrow	Ethanol \downarrow	Malonaldehyde \downarrow	Naphthalene \downarrow	Salicylic \downarrow	Toluene \downarrow	Uracil \downarrow
VE	0	1.247	0.364	0.448	0.735	0.483	0.785	0.480	0.575
	1	1.112	0.304	0.282	0.520	0.455	0.725	0.515	0.447
VP	0	1.087	0.358	0.300	0.880	0.517	0.788	0.540	0.675
	1	1.244	0.315	0.338	0.488	0.432	0.712	0.478	0.468

C.8.4. PaiNN as Backbone

Table 53. Results on 12 quantum mechanics prediction tasks from QM9, and the backbone model is PaiNN. We take 110K for training, 10K for validation, and 11K for testing. The evaluation is mean absolute error, and the best and the second best results are marked in **bold** and **bold**, respectively.

Pretraining	$\alpha \downarrow$	$\nabla \mathcal{E} \downarrow$	$\mathcal{E}_{\text{HOMO}} \downarrow$	$\mathcal{E}_{\text{LUMO}} \downarrow$	$\mu \downarrow$	$C_v \downarrow$	$G \downarrow$	$H \downarrow$	$R^2 \downarrow$	$U \downarrow$	$U_0 \downarrow$	ZPVE \downarrow
–	0.049	42.73	24.46	20.16	0.016	0.025	8.43	7.88	0.169	8.18	7.63	1.419
Distance Prediction	0.049	37.23	22.75	18.26	0.014	0.030	9.31	9.35	0.143	9.85	9.07	1.566
3D InfoGraph	0.047	44.25	24.06	18.54	0.015	0.052	8.81	7.97	0.143	8.68	8.08	1.416
GeoSSL-RR	0.046	41.20	23.93	19.36	0.016	0.025	8.32	8.17	0.174	7.99	8.20	1.438
GeoSSL-InfoNCE	0.045	39.29	23.23	18.40	0.015	0.024	8.34	8.37	0.127	7.45	8.34	1.356
GeoSSL-EBM-NCE	0.045	38.87	22.71	17.89	0.014	0.082	8.28	7.35	0.130	7.85	7.68	1.338
3D InfoMax	0.046	36.97	21.31	17.69	0.014	0.024	8.38	7.36	0.135	8.60	7.99	1.453
GraphMVP	0.044	36.03	20.71	17.02	0.014	0.024	8.31	7.36	0.132	7.57	7.34	1.337
GeoSSL-DDM-1L	0.045	36.13	20.59	17.26	0.014	0.024	9.45	8.43	0.128	8.88	8.16	1.380
GeoSSL-DDM	0.043	35.55	20.57	16.95	0.014	0.024	8.25	7.42	0.127	7.36	7.34	1.334
Uni-Mol	0.277	40.56	21.25	23.99	0.014	0.039	9.16	9.14	0.340	9.31	8.59	1.433
MoleculeSDE (VE)	0.044	34.67	20.14	17.05	0.013	0.023	7.64	7.05	0.139	6.88	6.79	1.273
MoleculeSDE (VP)	0.042	35.09	20.14	16.78	0.013	0.023	8.17	7.01	0.133	7.30	7.05	1.315

Table 54. Results on eight force prediction tasks from MD17, and the backbone model is PaiNN. We take 1K for training, 1K for validation, and 48K to 991K molecules for the test concerning different tasks. The evaluation is mean absolute error, and the best results are marked in **bold** and **bold**, respectively.

Pretraining	Aspirin \downarrow	Benzene \downarrow	Ethanol \downarrow	Malonaldehyde \downarrow	Naphthalene \downarrow	Salicylic \downarrow	Toluene \downarrow	Uracil \downarrow
–	0.572	0.053	0.230	0.338	0.132	0.288	0.141	0.201
Distance Prediction	0.480	0.053	0.200	0.296	0.131	0.265	0.171	0.168
3D InfoGraph	0.554	0.067	0.249	0.353	0.177	0.331	0.179	0.213
GeoSSL-RR	0.559	0.051	0.262	0.368	0.146	0.303	0.154	0.202
GeoSSL-InfoNCE	0.428	0.051	0.197	0.337	0.127	0.247	0.136	0.169
GeoSSL-EBM-NCE	0.435	0.048	0.198	0.295	0.143	0.245	0.132	0.172
3D InfoMax	0.479	0.052	0.220	0.344	0.138	0.267	0.155	0.174
GraphMVP	0.465	0.050	0.205	0.316	0.119	0.242	0.136	0.168
GeoSSL-DDM-1L	0.436	0.048	0.209	0.320	0.119	0.249	0.132	0.177
GeoSSL-DDM	0.427	0.047	0.188	0.313	0.120	0.240	0.129	0.167
Uni-Mol	0.487	0.048	0.217	0.329	0.151	0.299	0.141	0.182
MoleculeSDE (VE)	0.421	0.043	0.195	0.284	0.105	0.236	0.123	0.158
MoleculeSDE (VP)	0.443	0.045	0.191	0.301	0.131	0.261	0.140	0.159

C.8.5. Quality on Conformation Generation

Table 55. Results on conformation generation without FF optimization. The datasets are GEOM QM9 and GEOM Drugs.

Methods	GEOM QM9				GEOM Drugs			
	COV (%) \uparrow		MAT (\AA) \downarrow		COV (%) \uparrow		MAT (\AA) \downarrow	
	Mean	Median	Mean	Median	Mean	Median	Mean	Median
RDKit	83.26	90.78	0.3447	0.2935	60.91	65.70	1.2026	1.1252
CVGAE	0.09	0.00	1.6713	1.6088	0.00	0.00	3.0702	2.9937
GraphDG	73.33	84.21	0.4245	0.3973	8.27	0.00	1.9722	1.9845
CGCF	78.05	82.48	0.4219	0.3900	53.96	57.06	1.2487	1.2247
GeoMol	71.26	72.00	0.3731	0.3731	67.16	71.71	1.0875	1.0586
ConfGF	88.49	94.13	0.2673	0.2685	62.15	70.93	1.1629	1.1596
DMGC	96.23	99.26	0.2083	0.2014	96.52	100.00	0.7220	0.7161
GeoDiff	90.54	94.61	0.2090	0.1988	89.13	97.88	0.8629	0.8529
RMCF-R	–	–	–	–	82.25	90.77	0.839	0.789
RMCF-C	–	–	–	–	87.12	96.26	0.749	0.709
MoleculeSDE (ours)	92.37	97.21	0.2423	0.2356	85.42	99.49	0.9485	0.9041

C.9. Computational Cost on Pretraining

Table 56. Computational time on 17 pretraining algorithms. All the jobs are running on one single V100 GPU card. The four models in the first block are 2D SSL, the nine models in the second block are 3D SSL, and the four models in the last block are 2D-3D SSL.

Pretraining Algorithm	min / epoch
AttrMask	5.5 min/epoch
ContextPred	14 min/epoch
InfoGraph	6 min/epoch
MolCLR	10 min/epoch
Type Prediction	7.75 min/epoch
Distance Prediction	6.7 min/epoch
Angle Prediction	8 min/epoch
3D InfoGraph	7.5 min/epoch
RR	9.7 min/epoch
InfoNCE	10 min/epoch
EBM-NCE	10.8 min/epoch
GeoSSL-1L	11.2 min/epoch
GeoSSL	18 min/epoch
3D InfoMax	8.6 min/epoch
GraphMVP	11 min/epoch
MoleculeSDE (VE)	30 min/epoch
MoleculeSDE (VP)	30 min/epoch

Appendix D

Appendix for SGNN-EBM: Structured Multi-task Learning for Molecular Property Prediction

D.1. ChEMBL-STRING Dataset Generation

We propose ChEMBL-STRING, a multi-task learning dataset with explicit task relation for the molecular property prediction. This new dataset is built on the Large Scale Comparison (LSC) dataset [166], and we list the three main steps in Appendices D.1.1 to D.1.3.

D.1.1. Filtering molecules

Among 456,331 molecules in the LSC dataset, 969 are filtered out following the pipeline in [99]. Here we describe the detailed filtering process, and the molecules filtered out in each step.

- (1) Discard the **Nones** in the compound list.
- (2) Filter out the molecules with ≤ 2 non-H atoms.
- (3) Retain only the largest molecule in the SMILES string. *E.g.* if the compound is an organic hydrochloride, say $\text{CH}_3\text{NH}_3^+\text{Cl}^-$, we retain only the organic compound after removing HCl, in this case CH_3NH_2 .
- (4) Filter out molecules with molecular weight < 50 and 9 with molecular weight > 900 .

D.1.2. Querying the PPI scores

Then we obtain the PPI scores by querying the ChEMBL [168] and STRING [231] databases. The details are as follows:

- (1) The LSC dataset [166] gives the ChEMBL ID for each assay. We use the assay id to query the ChEMBL database by visiting [https://www.ebi.ac.uk/chembl/api/data/assay/\[assay_id\]](https://www.ebi.ac.uk/chembl/api/data/assay/[assay_id]) for target ID. We then query the ChEMBL database by visiting [https://www.ebi.ac.uk/chembl/api/data/target/\[target_id\]](https://www.ebi.ac.uk/chembl/api/data/target/[target_id]) for UniProt [35] information. We save all the UniProts related to each target in a list. We discard assays with no associated UniProt, and confirm that all remaining assays are targeting human proteins.
- (2) Next, we query the STRING database for the corresponding STRING ID. For each UniProt, we visit [https://string-db.org/api/xml/get_string_ids?identifiers=\[uniprot\]](https://string-db.org/api/xml/get_string_ids?identifiers=[uniprot]). We discard UniProts with no available StringIDs. The String ID list is then sent to <https://string-db.org/api/tsv-no-header/network> via a POST request to obtain the human PPI scores.

D.1.3. Constructing the Task Relation Graph

Finally, we calculate the edge weights w_{ij} , *i.e.*, task relation score, for task t_i and t_j in the task relation graph to be $\max\{\text{PPI}(s_i, s_j) : s_i \in S_i, s_j \in S_j\}$, where S_i denotes the protein set of task t_i . The resulting task relation graph has 1,310 nodes and 9,172 edges with non-zero weights. Note that 96% of the protein-targeted tasks only target a single protein, for which the relation score of these tasks is exactly the PPI score between their target proteins. We then densify the dataset via the following filtering process:

- (1) We filter out all isolated tasks.
- (2) We define a threshold τ and iteratively filter out molecules with number of labels below τ , tasks with number of labels below τ , and tasks with number of positive or negative labels below 10. We repeat this until no molecule or task is filtered out.

The statistics of the resulting ChEMBL-STRING dataset with three thresholds can be found at Table 14.

D.2. GIN for Molecule Embedding

The Graph Isomorphism Network (GIN) is proposed in [266]. It was originally proposed for the simple graph structured data, where each node has one discrete label and no extra edge information is provided. Here we adopt a customized GIN from a recent paper [99]. With this customized GIN as the base model, plus pre-training techniques, [99] can reach the state-of-the-art performance on several molecular property prediction tasks. Thus we adopt this customized GIN model in our work.

Following the notation in Section 3, each molecule is represented as a molecular graph, *i.e.*, $\mathbf{x} = (X, E)$, where X and E are feature matrices for atoms and bonds respectively. Suppose for one molecule, we have n atoms and m edges. The message passing function is defined as:

$$z_i^{(k+1)} = \text{MLP}_{\text{atom}}^{(k+1)}\left(z_i^{(k)} + \sum_{j \in \mathcal{N}(i)} \left(z_j^{(k)} + \text{MLP}_{\text{bond}}^{(k+1)}(E_{ij})\right)\right), \quad (\text{D.2.1})$$

where $z_0 = X$ and $\text{MLP}_{\text{atom}}^{(k+1)}$ and $\text{MLP}_{\text{bond}}^{(k+1)}$ are the $(l+1)$ -th MLP layers on the atom- and bond-level respectively. Repeating this for K times, and we can encode K -hop neighborhood information for each atom in the molecular data, and we take the last layer for each node/atom representation. The graph representation is the mean of the node representation, *i.e.*, the molecule representation in this paper:

$$z(\mathbf{x}) = \frac{1}{N} \sum_i z_i^{(K)} \quad (\text{D.2.2})$$

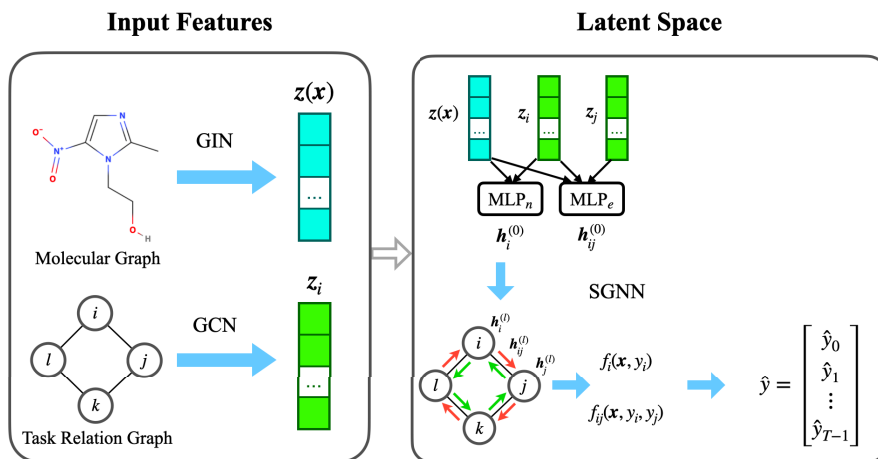


Figure 25. Pipeline of GNN. We first obtain molecule and task embedding via GIN and GCN. Then they are concatenated and passed through a GNN to better learn the task representation. The final prediction for each task is predicted independently on each node representation.

D.3. GCN for Task Embedding

We use graph convolutional network (GCN) [124] for the task embedding. For the i -th task, we first get its one-hot encoding and then pass it through an embedding layer, with the output denoted as $\mathbf{e}_i \in \mathbb{R}^{d_t \times 1}, \forall i \in \{0, 1, \dots, T-1\}$, where d_t is the task embedding dimension. $\mathbf{E} = \{\mathbf{e}_0, \mathbf{e}_1, \dots, \mathbf{e}_{T-1}\}^T \in \mathbb{R}^{T \times d_t}$ is the initial embedding matrix for T tasks. Then we pass \mathbf{E} through a GCN and the output embedding for the i -th task is $\mathbf{z}^{(i)} = \text{GCN}(\mathbf{E})_i, \forall i \in \{0, 1, \dots, T-1\}$.

D.4. SGNN for Modeling Latent Space

In this section, we give a detailed illustration of our proposed State GNN (SGNN) model in Section 5.2. The general pipeline is shown in Figure 25.

First let us quickly review the node- and edge-level inputs:

$$\begin{aligned} \mathbf{h}_i^{(0)}(\mathbf{x}) &= \text{MLP}_n^{(0)}(\mathbf{z}(\mathbf{x}) \oplus \mathbf{z}^{(i)}) \\ \mathbf{h}_{ij}^{(0)}(\mathbf{x}) &= \text{MLP}_e^{(0)}(\mathbf{z}(\mathbf{x}) \oplus \mathbf{z}^{(i)} \oplus \mathbf{z}^{(j)}), \end{aligned} \tag{D.4.1}$$

and as discussed in Section 5.2, the biggest difference between SGNN and the mainstream GNN models is that in SGNN, each node and each edge has two and four state respectively, where each state of a node/edge is the representation for the corresponding label. Recall that in this task relation graph, y_i is the label for the i -th task, and it has two values; similarly for each edge $\langle y_i, y_j \rangle$ has four labels with a simple combination. Thus the representations

for node label y_i and edge label $\langle y_i, y_j \rangle$ are as follows:

$$\begin{aligned} \mathbf{h}_i^{(0)}(\mathbf{x}, y_i) &= \mathbf{h}_i^{(0)}(\mathbf{x})[y_i] \\ \mathbf{h}_{ij}^{(0)}(\mathbf{x}, y_i, y_j) &= \mathbf{h}_{ij}^{(0)}(\mathbf{x})[y_i, y_j]. \end{aligned} \tag{D.4.2}$$

With the node and edge inputs, we can then define the message-passing propagation. Notice that here we are propagating on both the node- and edge-levels. Following the notations in Section 5.2, for the node-level propagation we have:

$$\begin{aligned} \mathbf{h}_i^{(l+1)}(\mathbf{x}, y_i) &= \text{MPNN}_n^{(l+1)}\left(\mathbf{h}_i^{(l)}(\mathbf{x}, y_i), \{\mathbf{h}_{ij}^{(l)}(\mathbf{x}, y_i, y_j) \mid \forall j, y_j\}\right) \\ &= \text{MLP}_n^{(l+1)}\left(\mathbf{h}_i^{(l)}(\mathbf{x}, y_i) + \sum_{j \in \mathbb{N}(i)} \sum_{y_j=0}^{C-1} \mathbf{h}_{ij}^{(l)}(\mathbf{x}, y_i, y_j)\right), \end{aligned} \tag{D.4.3}$$

and for the edge-level propagation, we have:

$$\begin{aligned} \mathbf{h}_{ij}^{(l+1)}(\mathbf{x}, y_i, y_j) &= \text{MPNN}_e^{(l+1)}\left(\mathbf{h}_i^{(l)}(\mathbf{x}, y_i), \mathbf{h}_j^{(l)}(\mathbf{x}, y_j), \mathbf{h}_{ij}^{(l)}(\mathbf{x}, y_i, y_j)\right) \\ &= \text{MLP}_e^{(l+1)}\left(\mathbf{h}_{ij}^{(l)}(\mathbf{x}, y_i, y_j) + \text{MLP}_a^{(l+1)}\left(\mathbf{h}_i^{(l)}(\mathbf{x}, y_i) + \mathbf{h}_j^{(l)}(\mathbf{x}, y_j)\right)\right), \end{aligned} \tag{D.4.4}$$

where $\text{MLP}_n^{(l+1)}(\cdot)$, $\text{MLP}_e^{(l+1)}(\cdot)$ and $\text{MLP}_a^{(l+1)}(\cdot)$ are MLP layers defined on the node-level, edge-level, and in the aggregation function from nodes to edges. All three MLP layers are mapping functions defined on $\mathbb{R}^d \rightarrow \mathbb{R}^d$.

D.5. Training Details

To train our proposed model, we use Adam for optimization with learning rate 1e-3, and the batch size is 32 for ChEMBL-STRING 10 (due to the memory issue) and 128 for ChEMBL-STRING 50 and ChEMBL-STRING 100. We train 200 epochs on ChEMBL-STRING 10 (within 36 hours) and 500 epochs on ChEMBL-STRING 50 and ChEMBL-STRING 100 (within 2 hours). The base graph neural network for molecule representation is GIN [266], and we follow the hyperparameter used in [99]. The base graph neural network for task embedding is GCN [124]. We have more detailed description of GIN and GCN in Appendices D.2 and D.3. The hyperparameter tuning for all baseline methods and SGNN base models in Appendix D.5.1.

D.5.1. Hyperparameter Tuning

We list the hyperparameters for baselines models and our proposed models in Table 57, including MTL, UW [119], GradNorm [31], Dynamic Weight Average (DWA) [157], and Loss-Balanced Task Weighting (LBTW) [147], SGNN in Section 5.2, SGNN-EBM in Section 5.4.

Table 57. Hyperparameters for baselines and our models.

Model	Hyperparameters	Values
MTL	Epochs	[100, 200]
UW	Epochs	[100, 200]
GradNorm	Epochs α	[100, 200] [0.1, 0.2, 0.5]
DWA	Epochs T	[100, 200] [0.2]
LBTW	Epochs α	[100, 200] [0.1, 0.2, 0.5]
SGNN	Epochs	[200, 500]
	d	[50, 100]
	# GIN Layer	[5]
	# GCN Layer	[0, 2]
	# SGNN Layer	[2]
SGNN-EBM	Epochs	[200, 500]
	Fixed-Noise Distribution Epochs	[200, 300, 400, 1000]
	d	[50, 100]
	# GIN Layer	[5]
	# GCN Layer	[0, 2]
	# SGNN Layer	[0, 2, 4]
	λ	[0.1, 1]

D.6. Noise Contrastive Estimation with Energy Tilting Term

Here we present the derivation of the training objective function of NCE learning with tilting term in Section 5.3. When applying the backbone model for noise distribution, *i.e.*, $p_n = q$, and adopting the self-normalization ($Z = 1$), the loss can be rewritten as:

$$\begin{aligned}
 \hat{\mathcal{L}}_{NCE} &= \mathbb{E}_{\mathbf{y} \sim p_n} \log \frac{p_n(\mathbf{y}|\mathbf{x})}{p_n(\mathbf{y}|\mathbf{x}) + p_\phi(\mathbf{y}|\mathbf{x})} + \mathbb{E}_{\mathbf{y} \sim p_{\text{data}}} \log \frac{p_\phi(\mathbf{y}|\mathbf{x})}{p_n(\mathbf{y}|\mathbf{x}) + p_\phi(\mathbf{y}|\mathbf{x})} \\
 &= \mathbb{E}_{\mathbf{y} \sim p_n} \log \frac{p_n(\mathbf{y}|\mathbf{x})}{p_n(\mathbf{y}|\mathbf{x}) + p_n(\mathbf{y}|\mathbf{x}) \exp(-E_\phi(\mathbf{x}, \mathbf{y}))} \\
 &\quad + \mathbb{E}_{\mathbf{y} \sim p_{\text{data}}} \log \frac{p_n(\mathbf{y}|\mathbf{x}) \exp(-E_\phi(\mathbf{x}, \mathbf{y}))}{p_n(\mathbf{y}|\mathbf{x}) + p_n(\mathbf{y}|\mathbf{x}) \exp(-E_\phi(\mathbf{x}, \mathbf{y}))} \\
 &= \mathbb{E}_{\mathbf{y} \sim p_n} \log \frac{1}{1 + \exp(-E_\phi(\mathbf{x}, \mathbf{y}))} + \mathbb{E}_{\mathbf{y} \sim p_{\text{data}}} \log \frac{1}{1 + \exp(E_\phi(\mathbf{x}, \mathbf{y}))}.
 \end{aligned} \tag{D.6.1}$$

For more detailed derivations, please check [153, 220].

Appendix E

Appendix for MoleculeSTM: Multi-modal Molecule Structure-text Model for Text-based Editing and Retrieval

E.1. Pretraining

E.1.1. PubChemSTM Construction

We construct a chemical structure-text pair dataset called PubChemSTM, which is extracted from the PubChem database [121]. Below we explain the key steps of the dataset construction.

- (1) We use the PUG View (a REST-style web service) to download the textual descriptions of molecules. It has in total of 290 pages, and each page is downloaded in XML format. For reference, an example page (the first page) can be found here. There is a “string” field in the XML data, and we treat it as the textual descriptions for molecules. After construction, we have 250K molecules (with unique PubChem ID) and 281K chemical structure-text pairs. Notice that each molecule can have multiple annotations from different resources.
 - Most of the molecule annotations start with the common name or the International Union of Pure and Applied Chemistry (IUPAC) name. We can either use the raw description (with a common name or IUPAC name) or replace it with the text template (*e.g.*, “This molecule is ...”).
 - Thus, we construct two versions of PubChemSTM datasets, PubChemSTM-raw and PubChemSTM-extracted, corresponding to using the raw annotation or replacing the molecule name with the text prompt, respectively. These two versions of PubChemSTM share the molecules, except for the molecule names.

- (2) We download the 326 SDF files from the PubChem FTP service. Each SDF file contains the structural information (*e.g.*, the SMILES string and molecular graph) for a batch of molecules.
- (3) We match the annotation and chemical structure for each molecule from the previous two steps using the PubChem ID, and most of the molecules from the first step contain the corresponding chemical structures from the SDF files. In specific, only 12 molecules failed to find the valid SMILES from SDF files, and we ignore these molecules.
- (4) Ultimately, following the above three steps will lead to a structure-text pair dataset with 281K pairs and 250K unique molecules. Note that the PubChem database [121] is updated online frequently, and the above numbers are collected in March 2022.

Pre-processing Details. There is one field in the PubChem database called "name", which includes either the common name or the IUPAC name for each molecule. Notice that the tokenization on IUPAC is nontrivial. Thus we carry out two versions to test its effect, *i.e.*, the PubChemSTM-raw and PubChemSTM-extracted. We find that there exist several patterns of textual descriptions in PubChemSTM-raw, which are further utilized to extract the cleaner version of molecule description as in PubChem-extract. A detailed illustration is given below:

- The most common pattern is that the molecule annotation starts with "XXX (name) is / are / was / were / appears / occurs / stands for / belongs to / exits ...". We manually extract this to obtain most of the molecule names and replace them with "This molecule ..." or "These molecules ...".
- **Extra word "Pure"**. Some molecule annotations start with "Pure xxx ..." and we remove the word "Pure".
- **Typos**. For example, the "Mercurycombines ..." should be "Mercury combines ...".

Dataset Examples. We provide four examples of the PubChemSTM-raw and PubChemSTM-extracted in Table 58.

Reproducibility. Because the PubChem database [121] has been updated online frequently, so we provide all the pre-processed datasets used in this work for reproducibility. In addition, the source codes for the above steps are also provided for future usage.

Comparison. As mentioned, we adopt a pretrained SciBERT model [13] and continue training on PubChemSTM. SciBERT is a BERT model specifically trained for scientific discovery. It randomly samples 1.14M papers from Semantic Scholar [3], where around 18% papers are from the computer science domain and 82% papers are from the broad biomedical domain. Its corpus has 3.17B tokens and the vocabulary size is 31K. Besides, SciBERT was trained on the full paper, not just the abstract. One potential issue is the vocabulary shift from the Semantic Scholar to PubChemSTM. Although we adapt the

Table 58. Examples on PubChemSTM. Here for the chemical structure, we only list the SMILES string, since the 2D topology graph can be obtained using the RDKit package.

PubChemSTM-raw	PubChemSTM-extracted
	SMILES: <chem>c1ccccc1</chem>
Benzene is a colorless liquid with a sweet odor. It evaporates into the air very quickly and dissolves slightly in water.	<i>This molecule is</i> a colorless liquid with a sweet odor. It evaporates into the air very quickly and dissolves slightly in water.
	SMILES: <chem>Oc1ccccc1</chem>
Phenol is both a manufactured chemical and a natural substance. It is a colorless-to-white solid when pure.	<i>This molecule is</i> both a manufactured chemical and a natural substance. It is a colorless-to-white solid when pure.
	SMILES: <chem>CC(=O)Oc1ccccc1C(=O)O</chem>
Acetylsalicylic acid appears as odorless white crystals or crystalline powder with a slightly bitter taste.	<i>This molecule appears</i> as odorless white crystals or crystalline powder with a slightly bitter taste.
	SMILES: <chem>CC1(C)SC2C(NC(=O)Cc3ccccc3)C(=O)N2C1C(=O)O</chem>
Benzylpenicillin is a penicillin in which the substituent at position 6 of the penam ring is a phenylacetamido group. It has a role as an antibacterial drug, an epitope and a drug allergen.	<i>This molecule is</i> a penicillin in which the substituent at position 6 of the penam ring is a phenylacetamido group. It has a role as an antibacterial drug, an epitope, and a drug allergen.

pretrained checkpoints from SciBERT (together with its vocabulary) in this work, we still want to carefully examine the vocabulary for the textual data.

Table 59. The vocabulary comparison.

Data Source	Tokenization Method	size of vocabulary	overlap with SciBERT
Semantic Scholar (used in SciBERT)	SciBERT tokenizer	31,090	-
PubChemSTM-raw	white space	315,704	7,635
	spaCy	114,976	719
	SciBERT tokenizer	18,320	18,320
PubChemSTM-extract	white space	100,877	7,562
	spaCy	27,519	691
	SciBERT tokenizer	17,442	17,442

In Table 59, we list the vocabulary size of PubChemSTM-raw and PubChemSTM-extract with three tokenization methods: using white space, spaCy [93], and the SciBERT tokenizer. We can observe that the difference between PubChemSTM-raw and PubChemSTM-extract using the SciBERT tokenizer is quite small, compared to the ones using white space and spaCy. Thus, we want to claim that vocabulary is also an important factor, and the SciBERT tokenizer has shown quite a stable tokenization effect. In the future, more comprehensive tokenization and vocabulary are required to push forwards this research line,

i.e., to enable the large language model for drug discovery. But it is beyond the scope of this paper and requires efforts from the entire community.

E.1.2. Architecture Details

We have two branches, the chemical structure branch f_c and the textual description branch f_t .

Chemical structure branch f_c . This work considers two types of chemical structures: the SMILES string views the molecule as a sequence and the 2D molecular graph takes the atoms and bonds as the nodes and edges, respectively. Then based on the chemical structures, we apply a deep learning encoder f_c to get a latent vector as molecule representation. Specifically, for the SMILES string, we take the encoder from MegaMolBART [106], which is pretrained on 500M molecules from ZINC database [224]. For the molecular graph, we take a pretrained graph isomorphism network (GIN) [266] using GraphMVP pretraining [153]. GraphMVP is doing a multi-view pretraining between the 2D topologies and 3D geometries on 250K conformations from GEOM dataset [9]. Thus, though we are not explicitly utilizing the 3D geometries, the state-of-the-art pretrained GIN models can implicitly encode such information.

Textual description branch f_t . The textual description branch provides a high-level description of the molecule’s functionality. We can view this branch as domain knowledge to strengthen the molecule representation. Such domain knowledge is in the form of natural language, and we use the BERT model [45] as the text encoder f_t . We further adapt the pretrained SciBERT [13], which was pretrained on the textual data from the chemical and biological domain.

Table 60. Model specifications. # parameters in each model.

Branch	Model	# parameters
Chemical structure	MegaMolBART	10,010,635
	GIN	1,885,206
Textual description	SciBERT	109,918,464

E.1.3. Pretraining Details

Pretraining Objective. For the MoleculeSTM pretraining, we apply contrastive learning. More concretely, we choose one of the EBM-NCE [153] and InfoNCE [181]. Both are essentially doing the same thing, yet EBM-NCE has been found to be more effective for graph-data [83, 153]. The objective for EBM-NCE is:

$$\begin{aligned} \mathcal{L} = & -\frac{1}{2} \left(\mathbb{E}_{\mathbf{x}_c, \mathbf{x}_t} [\log \sigma(E(\mathbf{x}_c, \mathbf{x}_t))] + \mathbb{E}_{\mathbf{x}_c, \mathbf{x}'_t} [\log(1 - \sigma(E(\mathbf{x}_c, \mathbf{x}'_t)))] \right) \\ & - \frac{1}{2} \left(\mathbb{E}_{\mathbf{x}_c, \mathbf{x}_t} [\log \sigma(E(\mathbf{x}_c, \mathbf{x}_t))] + \mathbb{E}_{\mathbf{x}'_c, \mathbf{x}_t} [\log(1 - \sigma(E(\mathbf{x}'_c, \mathbf{x}_t)))] \right), \end{aligned} \tag{E.1.1}$$

where \mathbf{x}_c and \mathbf{x}_t form the structure-text pair for each molecule, and $\mathbf{x}_{c'}$ and $\mathbf{x}_{t'}$ are the negative samples randomly sampled from the noise distribution, which we use the empirical data distribution. $E(\cdot)$ is the energy function with a flexible formulation, and we use the dot product on the jointly learned space, *i.e.*, $E(\mathbf{x}_c, \mathbf{x}_t) = \langle p_c \circ f_c(\mathbf{x}_c), p_t \circ f_t(\mathbf{x}_t) \rangle$. Similarly, we have the objective for InfoNCE as:

$$\mathcal{L} = -\frac{1}{2} \mathbb{E} \left[\log \frac{\exp(E(\mathbf{x}_c, \mathbf{x}_t))}{\exp(E(\mathbf{x}_c, \mathbf{x}_t)) + \sum_{\mathbf{x}_{t'}} \exp(E(\mathbf{x}_c, \mathbf{x}_{t'}))} + \log \frac{\exp(E(\mathbf{x}_c, \mathbf{x}_t))}{\exp(E(\mathbf{x}_c, \mathbf{x}_t)) + \sum_{\mathbf{x}_{c'}} \exp(E(\mathbf{x}_{c'}, \mathbf{x}_t))} \right]. \quad (\text{E.1.2})$$

Hyperparameters. We list the key hyperparameters used for MoleculeSTM pretraining with the SMILES string and 2D molecular graph as inputs, respectively.

Table 61. Hyperparameter specifications for MoleculeSTM pretraining.

Input	Hyperparameter	Value
SMILES string	epochs	{32}
	learning rate for text branch	{1e-4}
	learning rate for chemical structure branch	{1e-5, 3e-5}
	objective function	{EBM-NCE, InfoNCE}
2D molecular graph	epochs	{32}
	learning rate for text branch	{1e-4}
	learning rate for chemical structure branch	{1e-5, 3e-5}
	objective function	{EBM-NCE, InfoNCE}

E.2. Design Principles for Downstream Tasks

In this section, we discuss the key principles when designing specific tasks.

Applicable Evaluation. One of the biggest differences between the foundation model in the vision-language domain and our MoleculeSTM can be reflected in the evaluation. Most of the vision and language tasks can be viewed as art problems, *i.e.*, there does not exist a standard and exact solution that is applicable for evaluation. For instance, we can detect if the image is "a horse riding an astronaut" or "a panda making latte art" [201], but only visually not computationally, which prevents large-scale evaluation. This is not the case for drug discovery, because it is a scientific task, where the results (*e.g.*, properties of the output molecules in the editing task) can be evaluated exactly, either *in vitro* or *in silico*. Following this, the physical experiments are usually expensive and long-lasting, so in this work, we want to focus on tasks that are computationally feasible for evaluation.

Fuzzy Matching. Specifically for the molecule editing task, the text prompts should follow the "fuzzy matching" criterion because there could exist multiple output molecules. This is in contradiction with "exact matching", where the output molecules are deterministic. For example, for the functional group change, we can feed in the prompts like "change the third nitrogen in the ring to oxygen". This prompt is very explicit with an exact solution, and there exist rule-based chemistry tools in handling this problem perfectly. Thus, text-based editing cannot show its benefits in this track. Instead, text-based editing can provide more benefits in the fuzzy matching setting by wandering around the semantically meaningful directions in the latent space. This also reflects the *open vocabulary* attribute of the language model that we have been focusing on.

E.3. Downstream: Zero-shot Structure-text Retrieval

E.3.1. Dataset Construction

The DrugBank database [261] has many fields that can be interesting to explore drug discovery tasks. Here we extract three fields of each small molecule drug for the zero-shot retrieval task: the Description field, the Pharmacodynamics field, and the anatomical therapeutic chemical (ATC) field, as detailed below:

- **DrugBank-Description.** The Description field gives a high-level review of the drug’s chemical properties, history, and regulatory status.
- **DrugBank-Pharmacodynamics.** This illustrates how the drug modifies or affects the organism it is being used in. This field may include effects in the body that are desired and undesired (also known as the side effects).
- **DrugBank-ATC.** Anatomical therapeutic chemical (ATC) is a classification system that categorizes the molecule into different groups according to the organ or system on which they act and their therapeutic, pharmacological, and chemical properties.

We list the key steps in dataset construction as follows:

- (1) We download the full DrugBank database (in XML format) and small chemical structure files (in SDF format) from the website.
- (2) We parse the XML file, and extract the data with three fields: Description, Pharmacodynamics, and ATC.
- (3) We do the mapping from the extracted files to chemical structures in SDF files. For DrugBank-Description and DrugBank-Pharmacodynamics datasets, we exclude the molecules that have shown up in PubChemSTM, filtered with the canonical SMILES. Meanwhile, for DrugBank-ATC, we exclude the molecules satisfying the following two criteria simultaneously:

- **Chemical structure filtering** If the molecule with the same canonical SMILES has shown up in the PubChemSTM;
- **Textual data filtering** We first need to define a similarity between two textual data as in Equation (E.3.1), where $\text{text}_{\text{DrugBank}}$ and $\text{text}_{\text{PubChemSTM}}$ are the textual data for the same molecule from DrugBank and PubChemSTM, respectively, $\text{len}()$ is the length of textual data, and $\text{Levenshtein}()$ is the Levenshtein distance between two textual data. Thus, the second condition is: if the similarity between the DrugBank text and the PubChemSTM text is above a certain threshold (*e.g.*, 0.6).

Another detail is that, for DrugBank-ATC, there exist multiple ATC fields ($\text{text}_{\text{DrugBank}}$) for each small molecule. In PubChemSTM, there also exist multiple textual descriptions ($\text{text}_{\text{PubChemSTM}}$) for each molecule. Thus during the textual

data filtering step, for each shared molecule between DrugBank and PubChemSTM, we calculate the similarity for all the $\text{text}_{\text{DrugBank}}\text{-text}_{\text{PubChemSTM}}$ pairs, and exclude the molecule if there exists one pair with similarity above the threshold 0.6.

- (4) Some basic dataset statistics can be found in Table 62. Notice that ATC has many levels, and we are using level 5 for retrieval in this work.

$$\text{sim}(\text{text}_{\text{DrugBank}}, \text{text}_{\text{PubChemSTM}}) = 1 - \frac{\text{Levenshtein}(\text{text}_{\text{DrugBank}}, \text{text}_{\text{PubChemSTM}})}{\text{len}(\text{text}_{\text{DrugBank}})}. \quad (\text{E.3.1})$$

Table 62. Statistics on three fields in DrugBank. The filtering steps have been illustrated above.

Field	# molecules-text pairs molecule not in PubChemSTM	# molecules-text pairs molecule shared in PubChemSTM but text similarity below 0.6	total
DrugBank-Description	1,154	–	1,154
DrugBank-Pharmacodynamics	1,005	–	1,005
DrugBank-ATC	1,507	1,500	3,007

E.3.2. Experiments

For experiments, we introduce three baselines in the main body. As a proof-of-concept, we carry out another baseline called Random. For Random, both encoders (f_c and f_t) are randomly initialized. The zero-shot retrieval results on three datasets are shown in Tables 63 to 65.

Table 63. Accuracy (%) of DrugBank-Description T -choose-one retrieval.

	T	Given Chemical Structure			Given Text		
		4	10	20	4	10	20
SMILES	Random	24.59 ± 1.14	10.12 ± 1.38	4.97 ± 0.42	24.54 ± 0.97	9.97 ± 0.81	5.09 ± 0.37
	Frozen	25.07 ± 1.24	10.22 ± 1.19	5.12 ± 0.65	24.69 ± 1.87	10.20 ± 1.38	5.37 ± 1.15
	Similarity	36.35 ± 0.59	23.22 ± 0.58	16.40 ± 0.59	22.74 ± 0.24	10.31 ± 0.24	5.34 ± 0.24
	KV-PLM	73.80 ± 0.00	53.96 ± 0.29	40.07 ± 0.38	72.86 ± 0.00	52.55 ± 0.29	40.33 ± 0.00
	MoleculeSTM	97.50 ± 0.46	94.18 ± 0.46	91.12 ± 0.46	98.21 ± 0.00	94.54 ± 0.37	91.97 ± 0.46
Graph	Random	25.78 ± 1.43	10.71 ± 0.97	4.83 ± 1.00	24.98 ± 0.32	10.20 ± 0.40	4.80 ± 0.21
	Frozen	24.01 ± 1.34	9.39 ± 0.92	4.85 ± 0.52	24.00 ± 1.66	9.91 ± 0.71	5.07 ± 0.75
	Similarity	30.03 ± 0.38	13.63 ± 0.27	7.07 ± 0.10	24.81 ± 0.27	10.22 ± 0.24	4.74 ± 0.24
	KV-PLM	99.15 ± 0.00	97.19 ± 0.00	95.66 ± 0.00	99.05 ± 0.37	97.50 ± 0.46	95.71 ± 0.46
	MoleculeSTM	99.15 ± 0.00	97.19 ± 0.00	95.66 ± 0.00	99.05 ± 0.37	97.50 ± 0.46	95.71 ± 0.46

Table 64. Accuracy (%) of DrugBank-Pharmacodynamics T -choose-one retrieval.

	T	Given Chemical Structure			Given Text		
		4	10	20	4	10	20
SMILES	Random	24.49 ± 0.68	9.73 ± 0.34	5.14 ± 0.57	25.61 ± 0.62	10.10 ± 0.91	5.07 ± 0.69
	Frozen	25.47 ± 1.12	10.55 ± 0.75	5.48 ± 0.70	25.34 ± 0.41	9.86 ± 0.44	4.84 ± 0.26
	Similarity	27.85 ± 0.03	10.75 ± 0.02	5.67 ± 0.01	24.58 ± 0.03	11.25 ± 0.03	5.29 ± 0.02
	KV-PLM	68.38 ± 0.03	47.59 ± 0.03	36.54 ± 0.03	67.68 ± 0.03	48.00 ± 0.02	34.66 ± 0.02
	MoleculeSTM	88.07 ± 0.01	81.70 ± 0.02	75.94 ± 0.02	88.46 ± 0.01	81.01 ± 0.02	74.64 ± 0.03
Graph	Random	26.00 ± 0.37	9.65 ± 0.88	4.95 ± 0.36	25.11 ± 0.63	9.99 ± 0.62	4.82 ± 0.54
	Frozen	25.49 ± 1.82	10.19 ± 1.47	4.74 ± 0.56	25.55 ± 0.45	10.15 ± 0.77	4.88 ± 0.55
	Similarity	25.33 ± 0.27	9.89 ± 0.52	4.61 ± 0.08	25.28 ± 0.03	10.64 ± 0.02	5.47 ± 0.02
	KV-PLM	92.14 ± 0.02	86.27 ± 0.02	81.08 ± 0.05	91.44 ± 0.02	86.76 ± 0.03	81.68 ± 0.03
	MoleculeSTM	92.14 ± 0.02	86.27 ± 0.02	81.08 ± 0.05	91.44 ± 0.02	86.76 ± 0.03	81.68 ± 0.03

Table 65. Accuracy (%) of molecule-ATC T -choose-one retrieval.

	T	Given Chemical Structure			Given Text		
		4	10	20	4	10	20
SMILES	Random	25.03 ± 0.33	9.83 ± 0.19	4.80 ± 0.22	25.44 ± 1.21	10.03 ± 0.94	5.11 ± 0.79
	Frozen	25.05 ± 0.94	10.17 ± 0.63	4.99 ± 0.54	25.35 ± 0.78	10.32 ± 0.44	5.22 ± 0.34
	Similarity	30.03 ± 0.00	13.35 ± 0.02	7.53 ± 0.02	26.74 ± 0.03	11.01 ± 0.00	5.62 ± 0.00
	KV-PLM	60.94 ± 0.00	42.35 ± 0.00	30.32 ± 0.00	60.67 ± 0.00	40.19 ± 0.00	29.02 ± 0.00
	MoleculeSTM	70.84 ± 0.07	56.75 ± 0.05	46.12 ± 0.07	73.07 ± 0.03	58.19 ± 0.03	48.97 ± 0.06
Graph	Random	24.48 ± 0.66	9.97 ± 0.25	4.81 ± 0.34	25.48 ± 0.59	10.40 ± 0.37	5.38 ± 0.30
	Frozen	24.19 ± 0.77	10.24 ± 0.71	4.87 ± 0.47	24.95 ± 1.52	10.07 ± 0.80	5.06 ± 0.36
	Similarity	29.46 ± 0.00	12.34 ± 0.00	6.52 ± 0.00	25.78 ± 1.53	10.23 ± 0.70	5.06 ± 0.67
	KV-PLM	69.33 ± 0.03	54.83 ± 0.04	44.13 ± 0.05	71.81 ± 0.05	58.34 ± 0.07	47.58 ± 0.05
	MoleculeSTM	69.33 ± 0.03	54.83 ± 0.04	44.13 ± 0.05	71.81 ± 0.05	58.34 ± 0.07	47.58 ± 0.05

E.3.3. Ablation Study: Fixed Pretrained Encoders

In the main body, we conduct pretraining by adopting pretrained single-modality checkpoints, *i.e.*, the GraphMVP and MegaMolBART for f_c , and SciBERT for f_t . Then for MoleculeSTM pretraining, we use contrastive learning and update all the model parameters. Here we take an ablation study by only optimizing the projection layers to the joint space of the two branches (p_c, p_t) while keeping the two encoders (f_c, f_t) fixed. The results on the three datasets are shown in Tables 66 to 68.

Table 66. Accuracy (%) of DrugBank-Description T -choose-one retrieval.

	T	Given Chemical Structure			Given Text		
		4	10	20	4	10	20
SMILES	Random	24.59 ± 1.14	10.12 ± 1.38	4.97 ± 0.42	24.54 ± 0.97	9.97 ± 0.81	5.09 ± 0.37
	Frozen	25.07 ± 1.24	10.22 ± 1.19	5.12 ± 0.65	24.69 ± 1.87	10.20 ± 1.38	5.37 ± 1.15
	Similarity	36.35 ± 0.59	23.22 ± 0.58	16.40 ± 0.59	22.74 ± 0.24	10.31 ± 0.24	5.34 ± 0.24
	MoleculeSTM	47.64 ± 0.40	29.21 ± 0.47	19.69 ± 0.47	52.60 ± 0.46	32.24 ± 0.37	21.45 ± 0.37
Graph	Random	25.78 ± 1.43	10.71 ± 0.97	4.83 ± 1.00	24.98 ± 0.32	10.20 ± 0.40	4.80 ± 0.21
	Frozen	24.01 ± 1.34	9.39 ± 0.92	4.85 ± 0.52	24.00 ± 1.66	9.91 ± 0.71	5.07 ± 0.75
	Similarity	30.03 ± 0.38	13.63 ± 0.27	7.07 ± 0.10	24.81 ± 0.27	10.22 ± 0.24	4.74 ± 0.24
	MoleculeSTM	51.28 ± 0.00	31.99 ± 0.41	20.71 ± 0.47	55.27 ± 0.00	33.08 ± 0.00	21.77 ± 0.00

Table 67. Accuracy (%) of DrugBank-Pharmacodynamics T -choose-one retrieval.

	T	Given Chemical Structure			Given Text		
		4	10	20	4	10	20
SMILES	Random	24.49 ± 0.68	9.73 ± 0.34	5.14 ± 0.57	25.61 ± 0.62	10.10 ± 0.91	5.07 ± 0.69
	Frozen	25.47 ± 1.12	10.55 ± 0.75	5.48 ± 0.70	25.34 ± 0.41	9.86 ± 0.44	4.84 ± 0.26
	Similarity	27.85 ± 0.03	10.75 ± 0.02	5.67 ± 0.01	24.58 ± 0.03	11.25 ± 0.03	5.29 ± 0.02
	MoleculeSTM	46.43 ± 0.00	27.42 ± 0.47	18.24 ± 0.47	52.53 ± 0.41	30.53 ± 0.00	19.98 ± 0.00
Graph	Random	26.00 ± 0.37	9.65 ± 0.88	4.95 ± 0.36	25.11 ± 0.63	9.99 ± 0.62	4.82 ± 0.54
	Frozen	25.49 ± 1.82	10.19 ± 1.47	4.74 ± 0.56	25.55 ± 0.45	10.15 ± 0.77	4.88 ± 0.55
	Similarity	25.33 ± 0.27	9.89 ± 0.52	4.61 ± 0.08	25.28 ± 0.03	10.64 ± 0.02	5.47 ± 0.02
	MoleculeSTM	46.29 ± 0.03	27.18 ± 0.02	17.73 ± 0.02	50.95 ± 0.04	31.65 ± 0.03	23.00 ± 0.03

Table 68. Accuracy (%) of DrugBank-ATC T -choose-one retrieval.

	T	Given Chemical Structure			Given Text		
		4	10	20	4	10	20
SMILES	Random	25.03 ± 0.33	9.83 ± 0.19	4.80 ± 0.22	25.44 ± 1.21	10.03 ± 0.94	5.11 ± 0.79
	Frozen	25.05 ± 0.94	10.17 ± 0.63	4.99 ± 0.54	25.35 ± 0.78	10.32 ± 0.44	5.22 ± 0.34
	Similarity	30.03 ± 0.00	13.35 ± 0.02	7.53 ± 0.02	26.74 ± 0.03	11.01 ± 0.00	5.62 ± 0.00
	MoleculeSTM	43.41 ± 0.12	25.66 ± 0.06	15.69 ± 0.06	48.75 ± 0.11	29.44 ± 0.06	19.75 ± 0.03
Graph	Random	24.48 ± 0.66	9.97 ± 0.25	4.81 ± 0.34	25.48 ± 0.59	10.40 ± 0.37	5.38 ± 0.30
	Frozen	24.19 ± 0.77	10.24 ± 0.71	4.87 ± 0.47	24.95 ± 1.52	10.07 ± 0.80	5.06 ± 0.36
	Similarity	29.46 ± 0.00	12.34 ± 0.00	6.52 ± 0.00	25.78 ± 1.53	10.23 ± 0.70	5.06 ± 0.67
	MoleculeSTM	42.53 ± 0.07	24.34 ± 0.00	14.78 ± 0.03	48.91 ± 0.03	28.77 ± 0.07	19.28 ± 0.07

E.4. Downstream: Zero-shot Text-based Molecule Editing

Molecule editing or controllable molecule generation refers to changing the structures of the molecules based on a given and pretrained molecule generative model. In this work, with the help of a large language model in MoleculeSTM, we are able to do the zero-shot text-based molecule editing. First, we would like to list two key challenges, comparing the editing task between the vision domain and molecule domain, as follows:

- **Backbone generative model.** For domains in vision, the image controllable generation can be quite feasible based on StyleGAN [117], a well-disentangled backbone model. However, it is nontrivial for deep molecule generative models. A recent work GraphCG [152] has explored the disentanglement property of the graph-based controllable molecule generation methods, and the conclusion is that, even though the backbone generative models are not perfectly disentangled, there still exist methods for controllable generation on highly structured data like molecular graphs or point clouds. Meanwhile, developing a novel disentangled molecule generative model is out of the scope of this work, since the editing solution by MoleculeSTM is model-agnostic, and can be easily generalized to future models.
- **Evaluation.** Image controllable generation is an art problem, *i.e.*, it is subjective and can have multiple (or even infinitely many) answers. On the contrary, controllable molecule generation is a science problem, *i.e.*, it is objective and has only a few answers. This has been discussed in Appendix E.2.

E.4.1. Experiment Set-up

Implementation Details. Because most of the modules are fixed, we only need to learn the adaptor module and the optimized latent code w . The two key hyperparameters are the learning rate $\{1e-2, 1e-3\}$ and $\lambda \in \{1e1, 1e0, 1e-1, 1e-2, 1e-3\}$. As a fair comparison, for baselines, we take the form of $w = w_{in} + \alpha \cdot D$, where D is obtained using random, PCA and variance and $\lambda \in \{1.0, 1.5, 2.0, 2.5, 3.0\}$. For GS, we repeat the random sampling five times of each input molecule.

Next, we will conduct the zero-shot text-based molecule editing on four types of editing tasks, as well as three case study, as discussed below:

- Single-objective molecule editing in Appendix E.4.2 (eight tasks).
- Multi-objective molecule editing in Appendix E.4.3 (six tasks).
- Binding-affinity-based molecule editing in Appendix E.4.4 (six tasks).
- Drug relevance editing in Appendix E.4.5 (four tasks).
- Neighborhood searching for patent drug molecules in Appendix E.4.6 (three case studies).

Due to the page limit, we only show four multi-objective and four binding-affinity-based editing tasks in the main body. Here we show more comprehensive results.

E.4.2. Single-objective Molecule Editing

We first consider eight single-objective properties for molecule editing. As shown in the Methods section, the definitions of the satisfaction function and threshold Δ are based on each task specifically, as:

- We use LogP to evaluate the solubility and insolubility. We take 0 and 0.5 as the different thresholds.
- We use QED to evaluate the drug-likeness. We take 0 and 0.1 as the different thresholds.
- We use tPSA to evaluate the high and low permeability. We take 0 and 10 as the different thresholds.
- For the hydrogen bond acceptor (HBA) and hydrogen bond donor (HBD), we can directly count them in the molecules, and we use 0 and 1 as the different thresholds.

For Δ , it is the threshold that only difference above it can be viewed as a hit. So the larger Δ means a stricter editing criterion. Below we show both the quantitative and qualitative results on eight single-objective property molecule editing results.

Table 69. Results on eight single-objective molecule editing. The inputs are 200 molecules randomly sampled from ZINC, and the evaluation is the hit ratio of the property change. The latent optimization is text-based molecule editing with MoleculeSTM, with the SMILES string and the molecular graph, respectively.

	Δ	baseline				latent optimization	
		Random	PCA	High Variance	GS-Mutate	SMILES	Graph
This molecule is <i>soluble in water</i> .	0	35.33 \pm 1.31	33.80 \pm 3.63	33.52 \pm 3.75	52.00 \pm 0.41	61.87 \pm 2.67	67.86 \pm 3.46
	0.5	11.04 \pm 2.40	10.66 \pm 3.24	10.86 \pm 2.56	14.67 \pm 0.62	49.02 \pm 1.84	54.44 \pm 3.99
This molecule is <i>insoluble in water</i> .	0	43.36 \pm 3.06	39.36 \pm 2.55	42.89 \pm 2.36	47.50 \pm 0.41	52.71 \pm 1.67	64.79 \pm 2.76
	0.5	19.75 \pm 1.56	15.12 \pm 2.93	18.22 \pm 0.33	12.50 \pm 0.82	30.47 \pm 3.26	47.09 \pm 3.42
This molecule is <i>like a drug</i> .	0	38.06 \pm 2.57	33.99 \pm 3.72	36.20 \pm 4.34	28.00 \pm 0.71	36.52 \pm 2.46	39.97 \pm 4.32
	0.1	5.27 \pm 0.24	3.97 \pm 0.10	4.44 \pm 0.58	6.33 \pm 2.09	8.81 \pm 0.82	14.06 \pm 3.18
This molecule is <i>not like a drug</i> .	0	36.96 \pm 2.25	35.17 \pm 2.61	39.99 \pm 0.57	71.33 \pm 0.85	58.59 \pm 1.01	77.62 \pm 2.80
	0.1	6.16 \pm 1.87	5.26 \pm 0.95	7.56 \pm 0.29	27.67 \pm 3.79	37.56 \pm 1.76	54.22 \pm 3.12
This molecule has <i>high permeability</i> .	0	25.23 \pm 2.13	21.36 \pm 0.79	21.98 \pm 3.77	22.00 \pm 0.82	57.74 \pm 0.60	59.84 \pm 0.78
	10	17.41 \pm 1.43	14.52 \pm 0.80	14.66 \pm 2.13	6.17 \pm 0.62	47.51 \pm 1.88	50.42 \pm 2.73
This molecule has <i>low permeability</i> .	0	16.79 \pm 2.54	15.48 \pm 2.40	17.10 \pm 1.14	28.83 \pm 1.25	34.13 \pm 0.59	31.76 \pm 0.97
	10	11.02 \pm 0.71	10.62 \pm 1.86	12.01 \pm 1.01	15.17 \pm 1.03	26.48 \pm 0.97	19.76 \pm 1.31
This molecule has <i>more hydrogen bond acceptors</i> .	0	12.64 \pm 1.64	10.85 \pm 2.29	11.78 \pm 0.15	21.17 \pm 3.09	54.01 \pm 5.26	37.35 \pm 0.79
	1	0.69 \pm 0.01	0.90 \pm 0.84	0.67 \pm 0.01	1.83 \pm 0.47	27.33 \pm 2.62	16.13 \pm 2.87
This molecule has <i>more hydrogen bond donors</i> .	0	2.97 \pm 0.61	3.97 \pm 0.55	6.23 \pm 0.66	19.50 \pm 2.86	28.55 \pm 0.76	60.97 \pm 5.09
	1	0.00 \pm 0.00	0.00 \pm 0.00	0.00 \pm 0.00	1.33 \pm 0.24	7.69 \pm 0.56	32.35 \pm 2.57

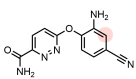
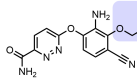
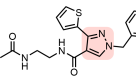
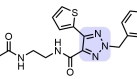
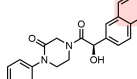
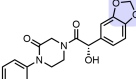
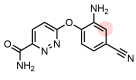
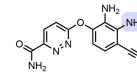
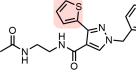
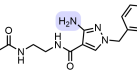
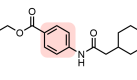
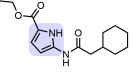
Table 70. Visualization of text-based editing on solubility, measured by the logarithm of the octanol-water partition coefficient (LogP) of the molecules. Generally, molecules with smaller LogP are more soluble in water. For generating molecules soluble in water, we can add polar components (*e.g.*, oxygens and nitrogens), remove hydrophobic moieties (*e.g.*, benzene and cyclohexane), or replace hydrophobic groups with polar functionalities in the input molecule. For generating molecules insoluble in water, we can make opposite modifications to the input molecule. The pink and blue regions highlight the modified structure in the input and output molecules, respectively.

Text Prompt: This molecule is <i>soluble in water</i> .					
Input Molecule	Output Molecule	Input Molecule	Output Molecule	Input Molecule	Output Molecule
 LogP: 3.66	 LogP: 3.05	 LogP: 3.72	 LogP: 2.56	 LogP: 4.25	 LogP: 2.76
Text Prompt: This molecule is <i>insoluble in water</i> .					
Input Molecule	Output Molecule	Input Molecule	Output Molecule	Input Molecule	Output Molecule
 LogP: 3.66	 LogP: 5.03	 LogP: -0.36	 LogP: 0.72	 LogP: 2.37	 LogP: 4.41

Table 71. Visualization of text-based editing on permeability, measured by the topological polar surface area (tPSA) of the molecules. Generally, molecules with smaller tPSA are more permeable. For generating molecules with high permeability, we can remove functional groups or heterocycles with high polarity from the input molecule, such as amides, sulfonamides, ureas, nitro groups, and nitrogen-containing arenes. For generating molecules with low permeability, we can make opposite modifications to the input molecule. The pink and blue regions highlight the modified structure in the input and output molecules, respectively.

Text Prompt: This molecule has <i>high permeability</i> .					
Input Molecule	Output Molecule	Input Molecule	Output Molecule	Input Molecule	Output Molecule
 tPSA: 104	 tPSA: 87	 tPSA: 96	 tPSA: 68	 tPSA: 76	 tPSA: 20
Text Prompt: This molecule has <i>low permeability</i> .					
Input Molecule	Output Molecule	Input Molecule	Output Molecule	Input Molecule	Output Molecule
 tPSA: 104	 tPSA: 116	 tPSA: 42	 tPSA: 67	 tPSA: 20	 tPSA: 46

Table 72. Visualization of text-based editing on hydrogen bond acceptors (HBA) and hydrogen bond donors (HBD). For generating molecules with more HBA, we can add heteroatoms to the input molecule such as oxygen, nitrogen, and sulfur, or replace existing groups with heteroatom-containing structural motifs. For generating molecules with more HBD, we can add heteroatoms that bear attached hydrogens, such as functional groups like amines, and heterocycles like pyrroles. The pink and blue regions highlight the modified structure in the input and output molecules, respectively.

Text Prompt: This molecule has <i>more hydrogen bond acceptors</i> .					
Input Molecule	Output Molecule	Input Molecule	Output Molecule	Input Molecule	Output Molecule
 HBA: 6	 HBA: 7	 HBA: 5	 HBA: 6	 HBA: 3	 HBA: 5
Text Prompt: This molecule has <i>more hydrogen bond donors</i> .					
Input Molecule	Output Molecule	Input Molecule	Output Molecule	Input Molecule	Output Molecule
 HBD: 2	 HBD: 3	 HBD: 2	 HBD: 3	 HBD: 1	 HBD: 2

E.4.3. Multi-objective Molecule Editing

We then consider six multi-objective properties for molecule editing. As shown in the Methods section, the definitions of the satisfaction function and threshold Δ are based on each task specifically. First, for each single-objective, we follow the evaluation metric in Appendix E.4.2, including the solubility, permeability, and the number of HBA and HBD. Then for the multi-objective evaluation, we consider two cases:

- The **simple** case with the loose thresholds, such as threshold 0 and 0 for solubility and permeability simultaneously.
- The **challenging** case with strict thresholds, such as threshold 0.5 and 1 for solubility and HBA/HBD simultaneously and threshold 0.5 and 10 for solubility and permeability simultaneously.

Then a successful hit needs to satisfy both conditions simultaneously. Below we show both the quantitative and qualitative results on six multi-objective property molecule editing results.

Table 73. Results on six multi-objective molecule editing. The inputs are 200 molecules randomly sampled from ZINC, and the evaluation is the hit ratio of the property change. The latent optimization is text-based molecule editing with MoleculeSTM, with the SMILES string and the molecular graph, respectively.

	Δ	baseline				latent optimization	
		Random	PCA	High Variance	GS-Mutate	SMILES	Graph
This molecule is <i>soluble in water</i> and has <i>more hydrogen bond acceptors</i> .	0 – 0	9.88 \pm 1.03	8.64 \pm 2.06	9.09 \pm 1.25	14.00 \pm 2.48	27.87 \pm 3.86	27.43 \pm 3.41
	0.5 – 1	0.23 \pm 0.33	0.45 \pm 0.64	0.22 \pm 0.31	0.67 \pm 0.62	8.80 \pm 0.04	11.10 \pm 1.80
This molecule is <i>insoluble in water</i> and has <i>more hydrogen bond acceptors</i> .	0 – 0	2.99 \pm 0.38	2.00 \pm 0.58	2.45 \pm 0.67	7.17 \pm 0.85	8.55 \pm 2.75	8.21 \pm 0.81
	0.5 – 1	0.45 \pm 0.32	0.00 \pm 0.00	0.22 \pm 0.31	0.17 \pm 0.24	2.93 \pm 0.30	0.00 \pm 0.00
This molecule is <i>soluble in water</i> and has <i>more hydrogen bond donors</i> .	0 – 0	2.28 \pm 1.15	2.23 \pm 1.16	4.44 \pm 0.58	13.83 \pm 2.95	33.51 \pm 4.08	49.23 \pm 1.71
	0.5 – 1	0.00 \pm 0.00	0.00 \pm 0.00	0.00 \pm 0.00	0.00 \pm 0.00	9.98 \pm 1.03	23.94 \pm 1.09
This molecule is <i>insoluble in water</i> and has <i>more hydrogen bond donors</i> .	0 – 0	0.69 \pm 0.58	1.96 \pm 0.87	1.79 \pm 0.66	5.67 \pm 0.62	17.03 \pm 2.75	14.42 \pm 3.43
	0.5 – 1	0.00 \pm 0.00	0.00 \pm 0.00	0.00 \pm 0.00	0.00 \pm 0.00	2.59 \pm 1.14	3.84 \pm 0.71
This molecule is <i>soluble in water</i> and has <i>high permeability</i> .	0 – 0	5.06 \pm 1.21	3.53 \pm 0.38	4.88 \pm 2.21	8.17 \pm 1.03	35.69 \pm 3.19	39.74 \pm 2.26
	0.5 – 10	1.16 \pm 0.68	0.67 \pm 0.55	0.66 \pm 0.54	0.00 \pm 0.00	19.15 \pm 0.73	22.66 \pm 1.90
This molecule is <i>soluble in water</i> and has <i>low permeability</i> .	0 – 0	12.17 \pm 1.05	10.43 \pm 2.88	13.08 \pm 2.28	19.83 \pm 2.46	44.35 \pm 0.68	30.87 \pm 0.62
	0.5 – 10	6.20 \pm 0.64	6.23 \pm 2.31	6.67 \pm 0.53	4.83 \pm 0.85	28.67 \pm 2.22	20.06 \pm 1.26

Table 74. Visualization of text-based editing on multi-objective (compositionality) properties: solubility and hydrogen bond donors (HBD), measured by LogP and number of HBD of the molecules. Molecules with more HBD are likely also soluble in water, such as replacing hydrophobic groups (benzene, thiophene, bromide, etc.) with polar groups or rings containing hydrogen-attached heteroatoms (alcohol, azaindole, carboxylic acid, etc.) in the input molecules. Nevertheless, we can add HBD to the input molecule while reducing its solubility, such as replacing high-polarity structural motifs (amide, lactone, etc.) with less hydrophilic HBD (indole, thiol, etc.) in the input molecules. The pink and blue regions highlight the modified structure in the input and output molecules, respectively.

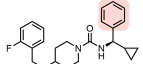
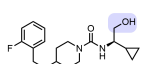
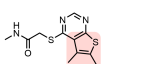
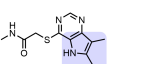
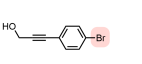
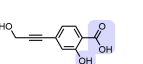
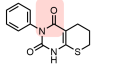
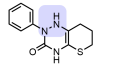
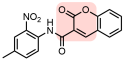
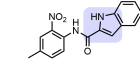
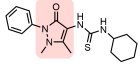
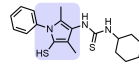
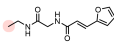
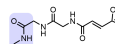
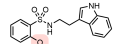
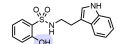
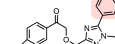
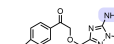
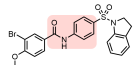
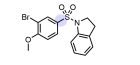
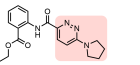
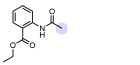
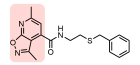
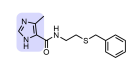
Text Prompt: This molecule is <i>soluble in water</i> and has <i>more hydrogen bond donors</i> .					
Input Molecule	Output Molecule	Input Molecule	Output Molecule	Input Molecule	Output Molecule
 LogP: 4.67, HBD: 1	 LogP: 2.29, HBD: 2	 LogP: 2.15, HBD: 1	 LogP: 1.41, HBD: 2	 LogP: 1.79, HBD: 1	 LogP: 0.43, HBD: 3
Text Prompt: This molecule is <i>insoluble in water</i> and has <i>more hydrogen bond donors</i> .					
Input Molecule	Output Molecule	Input Molecule	Output Molecule	Input Molecule	Output Molecule
 LogP: 1.56, HBD: 1	 LogP: 2.42, HBD: 2	 LogP: 3.26, HBD: 1	 LogP: 3.64, HBD: 2	 LogP: 3.10, HBD: 2	 LogP: 5.00, HBD: 3

Table 75. Visualization of text-based editing on multi-objective (compositionality) properties: solubility and permeability, measured by LogP and tPSA of the molecules. Molecules with low permeability are likely also soluble in water, such as adding polar functional groups (*e.g.*, amide, amine) and removing hydrocarbons (*e.g.*, methyl, phenyl) with regard to the input molecules. Nevertheless, we can increase both the solubility and permeability of the molecule, such as removing hydrocarbons and polar moieties simultaneously or reducing the size of the heterocycles (*e.g.*, [1,2]oxazolo[5,4-*b*]pyridine to imidazole) in the input molecules. The pink and blue regions highlight the modified structure in the input and output molecules, respectively.

Text Prompt: This molecule is <i>soluble in water</i> and has <i>low permeability</i> .					
Input Molecule	Output Molecule	Input Molecule	Output Molecule	Input Molecule	Output Molecule
 LogP: 0.55, tPSA: 71	 LogP: -0.34, tPSA: 100	 LogP: 2.70, tPSA: 71	 LogP: 2.39, tPSA: 82	 LogP: 3.70, tPSA: 93	 LogP: 1.62, tPSA: 119
Text Prompt: This molecule is <i>soluble in water</i> and has <i>high permeability</i> .					
Input Molecule	Output Molecule	Input Molecule	Output Molecule	Input Molecule	Output Molecule
 LogP: 4.46, tPSA: 76	 LogP: 3.21, tPSA: 47	 LogP: 2.51, tPSA: 84	 LogP: 1.82, tPSA: 55	 LogP: 3.50, tPSA: 68	 LogP: 2.38, tPSA: 58

E.4.4. Binding-affinity-based Molecule Editing

We further apply text-based editing on the binding affinity assays. In specific, we take six binding affinity tasks from ChEMBL [168]. Each assay has a textual description, as listed in Table 76.

Table 76. ChEMBL assay descriptions.

ChEMBL ID	Assay Description
1613777	This molecule is tested positive in <i>an assay that are inhibitors and substrates of an enzyme protein. It uses molecular oxygen inserting one oxygen atom into a substrate and reducing the second into a water molecule.</i>
1613797	This molecule is tested positive in <i>an assay for Anthrax Lethal, which acts as a protease that cleaves the N-terminal of most dual specificity mitogen-activated protein kinase kinases.</i>
2114713	This molecule is tested positive in <i>an assay for Activators of ClpP, which cleaves peptides in various proteins in a process that requires ATP hydrolysis and has a limited peptidase activity in the absence of ATP-binding subunits.</i>
1613838	This molecule is tested positive in <i>an assay for activators involved in the transport of proteins between the endosomes and the trans Golgi network.</i>
1614236	This molecule is <i>an inhibitor of a protein that prevents the establishment of the cellular antiviral state by inhibiting ubiquitination that triggers antiviral transduction signal and inhibits post-transcriptional processing of cellular pre-mRNA.</i>
1613903	This molecule is tested positive <i>in the high throughput screening assay to identify inhibitors of the SARS coronavirus 3C-like Protease, which cleaves the C-terminus of replicase polyprotein at 11 sites.</i>

For evaluation, we follow the Methods section. Recall that each binding affinity assay can correspond to molecules with positive and negative labels. Thus, we can train a classifier on these data points, and the satisfy criteria here is if the output molecules can have higher confidence than the input molecule, where the confidence is predicted using the classifier for each task. The pipeline can be found in Figure 26.

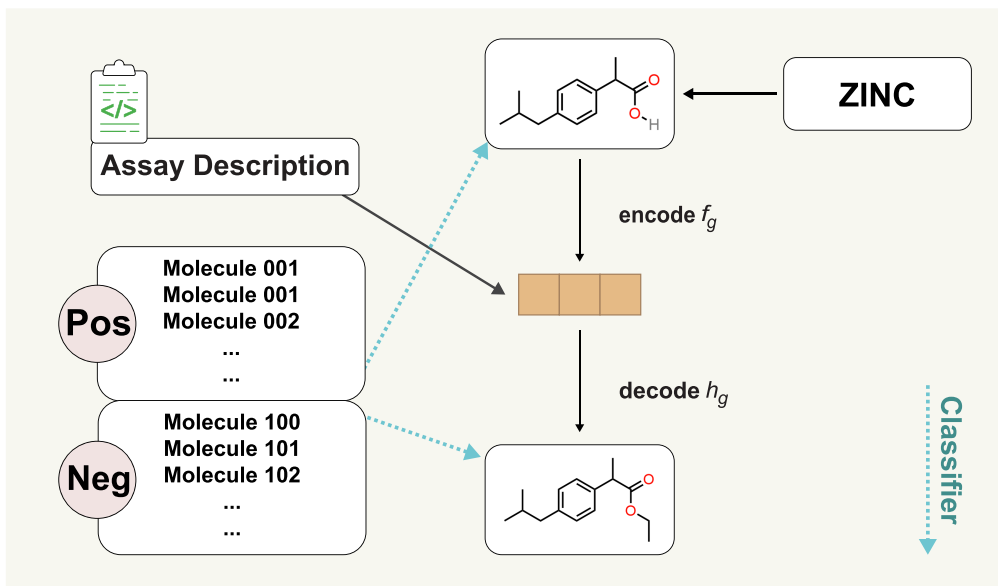


Figure 26. Pipeline for binding-affinity-based molecule editing. The input molecules are randomly sampled from ZINC, and the text prompt is the assay description. For evaluation, the small molecules for each assay are used to train a binary classifier, and two types of models (random forest and logistic regression) are considered.

The hit ratio results are shown in Table 77. Notice that to better prove the validity of our results, we train two classifiers for each assay: random forest (RF) and logistic regression (LR), with the fingerprint as featurization. the Δ is 0.

Table 77. Results on six ChEMBL assay editing. Each ChEMBL assay is a binary task and we train a classifier to obtain the confidence score of each molecule (input and output molecules). The inputs are 200 molecules randomly sampled from ZINC, and the evaluation is the hit ratio of the confidence change. The latent optimization is the text-based molecule editing with MoleculeSTM, with the SMILES string and the molecular graph, respectively.

ChEMBL ID		baseline				latent optimization	
		Random	PCA	High Variance	GS-Mutate	SMILES	Graph
1613777	RF	44.99 \pm 2.08	44.49 \pm 1.22	44.45 \pm 1.01	39.17 \pm 3.66	48.70 \pm 2.06	44.53 \pm 1.60
	LR	47.34 \pm 5.53	49.13 \pm 0.86	49.69 \pm 6.75	51.50 \pm 2.86	54.09 \pm 1.94	50.55 \pm 3.14
1613797	RF	44.76 \pm 2.18	46.25 \pm 0.97	46.92 \pm 3.34	46.67 \pm 1.55	55.03 \pm 2.23	49.03 \pm 0.03
	LR	48.40 \pm 3.71	49.92 \pm 4.31	48.67 \pm 1.64	49.17 \pm 3.01	57.98 \pm 3.34	54.95 \pm 3.74
2114713	RF	39.87 \pm 2.32	42.91 \pm 2.64	42.19 \pm 3.68	41.33 \pm 1.25	49.20 \pm 2.11	60.93 \pm 2.53
	LR	51.39 \pm 1.15	52.62 \pm 1.64	52.24 \pm 1.07	50.50 \pm 1.47	56.93 \pm 3.67	58.77 \pm 2.41
1613838	RF	44.49 \pm 1.48	44.71 \pm 1.80	45.30 \pm 2.47	36.00 \pm 2.68	43.94 \pm 3.75	49.13 \pm 2.52
	LR	50.22 \pm 4.23	49.73 \pm 2.33	44.69 \pm 2.41	41.33 \pm 3.17	47.50 \pm 2.28	56.13 \pm 1.50
1614236	RF	41.33 \pm 3.59	42.28 \pm 1.91	42.85 \pm 2.88	45.33 \pm 1.65	57.90 \pm 2.39	35.71 \pm 4.19
	LR	46.57 \pm 0.51	49.34 \pm 1.80	50.62 \pm 3.86	56.00 \pm 1.08	65.78 \pm 5.67	46.36 \pm 2.53
1613903	RF	44.28 \pm 0.77	43.83 \pm 2.65	42.00 \pm 3.19	46.17 \pm 0.85	56.82 \pm 3.96	58.70 \pm 1.43
	LR	53.94 \pm 3.30	48.63 \pm 4.49	56.19 \pm 2.51	56.33 \pm 0.94	58.31 \pm 2.98	64.64 \pm 5.23

Then we add docking for visualization in Figure 27. We choose the ChEMBL 1613777 with the available PDB structure. In specific, we first extract the output molecules using MoleculeSTM with confidence (RF and LR) higher than the ones generated with baselines. Then we run the molecular docking software for the results. The details of docking settings are listed below.

- We use Merck molecular force field (MMFF) [81] provided in RDKit [129] to embed (generate) 3D conformers for each molecule. The dielectric constant is set to be 80 and the maximum iteration of optimization is 1000 for MMFF, and the up-to-5 conformers from each molecule are used for further analysis.
- For the binding target, we consider assay P450 (CYP) 2C19 [197] (ChEMBL id: 1613777) and select the corresponding crystal structure available in the Protein Data Bank (PDB) (PDB id: 4GQS). Further, we take chain A for docking running. Later for the binding, the binding pockets are aligned with the original ligand in the crystal structure of PDB complexes: the center is set to (-81.48, 16.55, -41.6), and the box is (20.0, 23.0, 25.0).
- Then we take a preprocessing step to complement the hydrogen atoms and add partial charges. We utilize meeko v0.3.3 for small molecules and AutoDock Flexible Receptor (ADFR) suite v1.2 for proteins.
- For docking, we use AutoDock Vina v1.2.3 [238]. Each molecule conformer is docked with *exhaustiveness* being 32, and the pose with the best (lowest) docking score is picked and used for visualization. For visualization, we use UCSF Chimera.

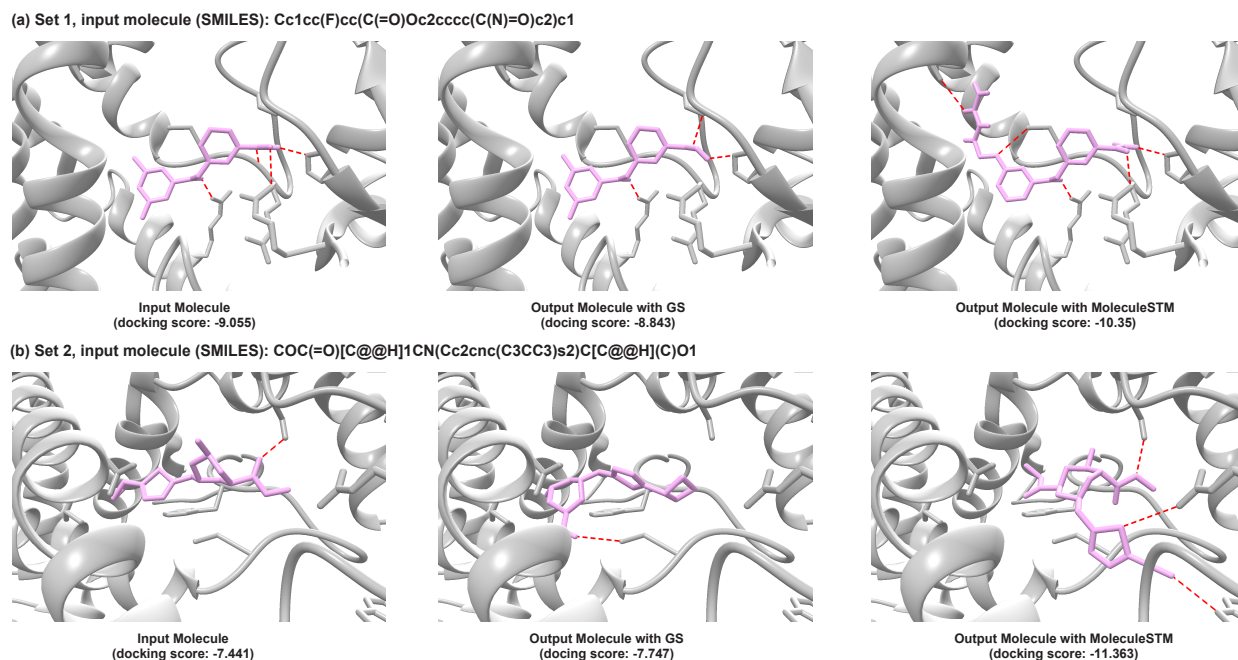


Figure 27. Two sets of docking visualization for binding-affinity-based molecule editing. The text prompt is from ChEMBL 1613777 (“This molecule is tested positive in *an assay that are inhibitors and substrates of an enzyme protein. It uses molecular oxygen inserting one oxygen atom into a substrate, and reducing the second into a water molecule.*”). For visualization, the input molecule and output molecules with GS and MoleculeSTM are displayed. It is observed that MoleculeSTM can generate molecules with the lowest docking scores (with the most Hydrogen bonds, and marked in red dashed lines). In set 1 (a), the output molecules are sharing the same molecule scaffold. In set 2 (b), the motif of the output molecule using MoleculeSTM also changes.

E.4.5. Drug Relevance Editing

As a proof-of-concept, we further take four editing tasks on common drug editing. The text prompts used here are to make the input molecules look like an existing drug, *e.g.*, “This molecule looks like *Penicillin*.” Following the Methods section, the satisfy function used is the Tanimoto similarity, and the threshold Δ takes the value of 0 and 0.05.

Table 78. Results on four common drug molecule editing. The inputs are 200 molecules randomly sampled from ZINC, and the evaluation is the hit ratio on the increase of the Tanimoto similarity with the common drug. The latent optimization is text-based molecule editing with MoleculeSTM, with the SMILES string and the molecular graph, respectively.

	Δ	baseline				latent optimization	
		Random	PCA	High Variance	GS-Mutate	SMILES	Graph
This molecule <i>looks like Penicillin</i> .	0	43.61 \pm 2.23	46.51 \pm 3.02	44.42 \pm 3.56	28.67 \pm 0.94	58.13 \pm 0.97	50.91 \pm 2.80
	0.05	0.69 \pm 0.55	0.23 \pm 0.32	0.89 \pm 0.30	0.67 \pm 0.62	11.01 \pm 0.58	3.64 \pm 0.57
This molecule <i>looks like Aspirin</i> .	0	43.82 \pm 1.41	43.12 \pm 5.35	44.63 \pm 3.33	25.00 \pm 2.16	40.13 \pm 1.33	54.05 \pm 3.58
	0.05	2.99 \pm 0.38	3.08 \pm 0.82	2.45 \pm 0.33	0.33 \pm 0.47	4.28 \pm 1.22	10.84 \pm 1.26
This molecule <i>looks like Caffeine</i> .	0	42.71 \pm 3.16	40.33 \pm 0.71	40.64 \pm 3.89	26.17 \pm 1.31	46.08 \pm 3.81	51.01 \pm 1.22
	0.05	0.69 \pm 0.01	0.23 \pm 0.32	0.44 \pm 0.31	0.33 \pm 0.24	1.61 \pm 0.67	0.61 \pm 0.01
This molecule <i>looks like Dopamine</i> .	0	42.00 \pm 3.08	42.50 \pm 2.12	41.33 \pm 2.86	30.50 \pm 1.63	47.00 \pm 4.11	55.50 \pm 2.73
	0.05	0.00 \pm 0.00	0.44 \pm 0.31	0.22 \pm 0.31	0.83 \pm 0.24	2.30 \pm 0.44	6.24 \pm 0.56

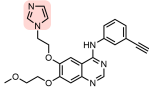
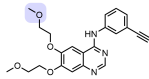
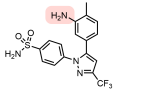
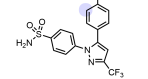
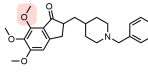
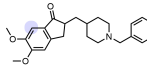
E.4.6. Case Studies on Neighborhood Searching for Patent Drug Molecules

To demonstrate the utility of text-based molecule editing, we show three case studies of generating approved drugs from their analogs. Lead optimization is a critical phase of drug discovery in which closely related compounds are made based on the lead molecule, aiming to improve its efficacy and DMPK (drug metabolism and pharmacokinetics) properties and ultimately identifying a drug candidate [103]. A text prompt calling for greater drug-like properties will thus be informative towards improving on deficiencies in the lead molecule and accelerating drug discovery research.

In specific here, the input molecules are the patented analogs of each approved drug molecule, and the input text prompt is single-objective, like the ones in Appendix E.4.2. The goal here is to check if the approved drugs can be successfully generated as the output molecules, with the structural changes consistent with the property improvement reflected in the text prompt. For example, in Table 79 (a), Erlotinib is successfully generated from an analog by replacing an imidazole substituent to a methoxy group [205]. This change reflects a tPSA drop from 83 to 75, consistent with the text prompt indicating a higher permeability. Table 79 (b) generates Celecoxib from its amino-substituted derivative [232], where the removal of the amino group yields a greater intestinal permeability of the molecule leading to higher bioavailability. Bioavailability is the fraction of a drug molecule that reaches

the systemic circulation, a key factor for oral drug absorption [39]. Finally, Table 79 (c) illustrates how potential metabolic liabilities in a molecule can be addressed via text-based editing. A text calling for a metabolically stable molecule successfully turns a trimethoxy arene to a dimethoxy arene in Donepezil [225], where the former represents an electron-rich aromatic compound known to undergo oxidative phase I metabolisms [78].

Table 79. Visualization on three single-objective molecule editing on drug analogs that generates approved drugs based on the text prompt. The pink and blue regions highlight the modified structure in the input and output molecules, respectively.

(a) Prompt: This molecule has <i>high permeability</i> .		(b) Prompt: This molecule has <i>high bioavailability</i> .		(c) Prompt: This molecule is <i>metabolically stable</i> .	
Input Molecule	Output Molecule	Input Molecule	Output Molecule	Input Molecule	Output Molecule
 CAS: 183320-43-6	 Tarceva (Erlotinib)	 CAS: 170570-28-2	 Celebrex (Celecoxib)	 CAS: 120013-52-7	 Aricept (Donepezil)

E.5. Downstream: Molecular Property Prediction

In this section, we review two main categories of datasets used for molecular property prediction downstream tasks from MoleculeNet and molecule benchmarking works [250, 263].
Molecular Property: Pharmacology. The Blood-Brain Barrier Penetration (BBBP) [165] dataset measures whether a molecule will penetrate the central nervous system. All three toxicity-related datasets, Tox21 [237], ToxCast [263], and ClinTox [66] are related to the toxicity of molecular compounds. The Side Effect Resource (SIDER) [128] dataset stores the adverse drug reactions on a marketed drug database.

Molecular Property: Biophysics. Maximum Unbiased Validation (MUV) [198] is another sub-database from PCBA, and is obtained by applying a refined nearest neighbor analysis. HIV is from the Drug Therapeutics Program (DTP) AIDS Antiviral Screen [277], and it aims at predicting inhibit HIV replication. BACE measures the binding results for a set of inhibitors of β -secretase 1 (BACE-1) and is gathered in MoleculeNet [263].

Table 80. Summary for the molecule chemical datasets.

Dataset	Task	# Tasks	# Molecules
BBBP	Classification	1	2,039
Tox21	Classification	12	7,831
ToxCast	Classification	617	8,576
Sider	Classification	27	1,427
ClinTox	Classification	2	1,478
MUV	Classification	17	93,087
HIV	Classification	1	41,127
Bace	Classification	1	1,513

For data splitting, we adopt the scaffold splitting [263]. Scaffold measures the skeleton structure of molecules, and scaffold splitting means we will put the molecules with more common scaffolds into training, and the rest into validation and test, so as to mimic the out-of-distribution (OOD) setting. The OOD setting is more common in real scenarios and thus is preferred to test the pretrained molecule representation power.

Implementation Details. For the SMILES string, we use MegaMolBART [106] as the backbone Transformer model. For the molecular graph, we use the same backbone GIN model, and we use rich features (as used for the regression tasks in GraphMVP [153]). We list the main hyperparameters below.

Table 81. Hyperparameter specifications for molecular property prediction.

	Hyperparameter	Value
Pretraining Baseline	epochs	{100}
	learning rate	{1e-3}
	weight decay	{0}
Downstream	epochs	{100}
	learning rate	{1e-3, 5e-4}
	weight decay	{0}